

Sintaxa:

```
<HTML>
  <HEAD>
    <TITLE> Titlul ferestrei </TITLE>
  </HEAD>
  <BODY>
    Conținut pagină
  </BODY>
</HTML>
```

Culori

Sistemul general de definire a culorilor RGB (**R**ed=rosu, **G**reen=verde, **B**lue=albastru)
- cod 6 cifre hexa #FF0000 - rosu, #00FF00 - verde, #0000FF - albastru sau cod cifre in zecimal (de exemplu #63FF80 va avea codul zecimal 99, 255, 128).

Cele 16 culori standard:

#00FFFF - aqua
#000000 - black - negru
#0000FF - blue - albastru
#FF00FF - fuchsia
#808080 - gray
#008000 - green - verde
#00FF00 - lime
#800000 - maroon
#000080 - navy
#808000 - olive
#800080 - purple
#FF0000 - red - rosu
#C0C0C0 - silver
#008080 - teal
#FFFF00 - yellow - galben
#FFFFFF - white - alb

Pentru acestea se poate scrie numele culorii. Pe langa acestea, mai sunt 100 de nuante de gri (gray) numerotate astfel:

gray 1, gray 2, ... , gray 100

bgcolor - culoarea fundalului

text - culoarea textului

link - culoarea linkurilor nevizitate (implicit albastru)

vlink - culoarea linkurilor vizitate (implicit violet)

alink - culoarea linkului activ (implicit rosu)

Exemplu: <BODY vlink="magenta"> - seteaza culoarea magenta pentru linkurile vizitate.

Numarul posibil de culori e 16.777.000 furnizate de codul RGB.

Culori sigure sunt cele cu 00, 33, 66, 99, CC si FF. De exemplu, un monitor setat sa afiseze 256 de culori va incerca sa afiseze orice culoare in afara acestui set cu cea mai apropiata culoare din setul de 256 (fenomen numit "dithering").

Caractere (fonturi)

size - dimensiune text

color - culoarea textului

face - tipul de font (arial, tahoma, helvetica, Times New Roman, Courier, Verdana)

Se pot preciza mai multe fonturi despartite prin virgula asa:

Pentru întregul document (pagini Web) se poate folosi eticheta <BASEFONT> care e bine să fie pusă imediat după eticheta <BODY>
Obs. Ea nu este o eticheta container.

Alte etichete:

<BIG> text </BIG> face fonturile mai mari
<SMALL> text </SMALL> face fonturile mai mici
 - bold (ingrosat)
<I> </I> și <EM - italic/emphasized (inclinat)
<U> </U> - underlined (subliniat)
<S> </S> sau <STRIKE> </STRIKE> - tăiat
 - superscript (exponent)
 - subscript (indice)
<CITE> </CITE> - citation (citată)
<CODE> </CODE> - code (pentru a insera secvențe de cod scrise într-un limbaj de programare)
<DFN> </DFN> - definition (pentru definiția unor termeni)
<KBD> </KBD> - keyboard (pentru a indica un text care urmează să fie introdus de la tastatură)
<SAMP> </SAMP> - sample (pentru a insera o mostră de text)
<TT> </TT> - teletype (pentru a afișa textul inclus între etichete cu fonturi monospațiate - adică lui "i" i se rezervă același număr de pixeli pe ecran ca și pentru "w").

Formatare text

 - trece la un rând nou (cu atributul clear <BR clear="left"> sau <BR clear="right"> rândul următor va începe de sub o eventuală imagine, tabel etc. și aliniat stânga sau dreapta)
<H1> </H1> ... <H6> </H6> cu atributul align – pentru titluri (Headings) ~ 1 – cel mai mare și 6 - cel mai mic.
<P> </P> - paragraf (cu atributul align)
<PRE> </PRE> - scrie textul așa cum a fost formatat în documentul sursă
<CENTER> </CENTER> - centrare text
<NOBR> </NOBR> - afișare text pe o singură linie obligatoriu (împiedică browserul să limiteze lungimea liniei la redimensionarea ferestrei)
<DIV> </DIV> - divizarea unui document în secțiuni distincte (are atributul align pentru aliniere și atributul nowrap – care interzice întreruperea rândurilor de către browser)
<HR> - linii orizontale (Horizontal Rule) cu atributele: align, width, size și color
<ADDRESS> </ADDRESS> - pentru inserare adresa poștală
<BLOCKQUOTE> </BLOCKQUOTE> - pentru inserare citată pe mai multe linii (prin indentarea sa)
<CITE> </CITE> - pentru inserare citată în cadrul unei linii (scriere cu caractere italice)
&cod; - comandă ampersand pentru inserarea caracterelor speciale și se face astfel:

Denumire caracter	Reprezentare	Comanda &
Spatiu liber		
Copyright	©	©
Trademark	™	™
Registered	®	® sau ®
Mai mic decât	<	<
Mai mare decât	>	>
Ampersand	&	&
Ghilimele	"	"
Cent	¢	¢
Un sfert	¼	¼
O jumătate	½	½
Trei sferturi	¾	¾
Grade	°	°

Legături (links)

 text sau imagini

href provine de la *Hipertext Reference*

Sintaxa generala URL:

schema://server_gazda:port/calea_catre_fisier

schema – reprezinta unul din protocoalele de transfer (http, ftp, Gopher, Telnet, etc.)

server_gazda – poate fi numele serverului sau IP-ul sau

port – reprezinta numarul portului de comunicatie prin intermediul caruia browserul se conecteaza la server, fiecare servind unui alt tip de comunicatie (portul 80 este cel standard pentru http si acesta nu se mai scrie, fiind implicit).

calea_catre_fisier – reprezinta localizarea ierarhica a fisierului in sistemul de directoare de pe server (aceasta consta in unul sau mai multe nume despartite prin caracterul *slash „/”*)

Observatie: In cazul cand fisierul se afla pe calculatorul personal, se foloseste un URL de forma:

file://server/calea_catre_fisier

unde protocolul este **file** iar numele computerului personal poate fi inlocuit prin cuvantul generic **localhost**

Acest nume poate fi omis, dar in acest caz este necesara prezenta unui caracter „/” suplimentar.

In functie de localizarea tintei (a resursei referita de legatura) pot exista ancore:

- **in cadrul aceleiasi pagini** (pentru a facilita navigarea in text de mari dimensiuni):

1. se plaseaza in text un punct de legatura cu ajutorul atributului name, astfel:

** **

Obs: nume_ancora este *case sensitive*.

2. se realizeaza legatura propriu-zisa folosind atributul href cu semnul # plasat inaintea numelui ancorei pentru a indica browserului o legatura interna astfel:

** text explicativ **

- **legatura catre o pagina aflata in acelasi director (folder):**

** text explicativ **

Obs: nume_fisier este *case sensitive*.

- **legatura catre o pagina aflata in alt director (folder):**

** text explicativ **

Unde **cale (path)** este calea relativa sau absoluta catre folderul in care se afla fisierul.

Obs: pat si nume_fisier sunt *case sensitive*.

- **legatura catre o pagina externa:**

** text explicativ **

- **legatura la aplicatia pentru expediere e-mail:**

ceea ce determina lansarea in executie a aplicatiei de expediere e-mail de catre cel ce viziteaza pagina si da cloc pe aceasta legatura. Programul implicit de posta electronica al utilizatorului se deschide cu camul adresa destinatar completata „To:adresa_e_mail” .

- **legatura catre un fisier de alt tip decat html**, ceea ce declaseaza procesul de transfer sau descarcare (download) urmand ca dupa transfer, utilizatorul sa deschida fisierul cu un program adecvat.

Observatie: Pentru ca noua pagina html sa se deschida intr-o alta fereastră, se utilizeaza atributul target cu valoarea „_blank” astfel:

** text explicativ **

Pentru a se deschide in aceeasi fereastră cu pagina sursa, atributului target i se asociaza valoarea „_self” astfel:

** text explicativ **

Imagini si elemente multimedia

- imagini *in-line* (direct in pagina)

Atribute:

border – a carui valoare reprezinta numarul de pixeli pentru grosimea bordurii (chenarului) din jurul imaginii (absenta atributului sau border="0" face ca chenarul sa nu fie prezent).

alt – stabileste un text explicativ afisat in spatiul in care va fi afisata imaginea. Acest mesaj alternativ va fi afisat in cazul in care nu se poate afisa imaginea si atunci cand mentinem cursorul mouseului deasupra

imaginii. El apare si inaintea incarcarii imaginii utilizatorii afland ce se va incarca acolo. Browserele recunosc fisiere grafice JPEG (Joint Photographic Experts Group) si GIF (Graphics Interchange Format).

width – stabileste latimea imaginii

height – stabileste inaltimea imaginii

align – aliniere text si imagine (**left** - imaginea in stanga si textul in dreapta sa, **right** - imaginea in dreapta si textul in stanga sa, **top** – partea de sus a imaginii se aliniaza cu prima linie a textului ce precede imaginea, **middle** - mijlocul imaginii se aliniaza cu prima linie a textului ce precede imaginea, **bottom** - partea de jos a imaginii se aliniaza cu prima linie a textului). Valorile left, right si bottom permit ca textul sa fie dispus in jurul imaginii, in vreme ce top si middle nu permit acest lucru.

Obs. Atributul align nu poate lua valoare center, imaginea putand fi centrata doar daca este izolata de textul care o inconjoara fie prin eticheta <CENTER> fie prin eticheta <DIV>, acestea avand setat atributul align la center.

hspace – precizeaza distanta in pixeli pe orizontala dintre imagine si restul elementelor din pagina.

vspace – precizeaza distanta in pixeli pe verticala dintre imagine si restul elementelor din pagina.

Observatie:

1. Este indicat a se folosi atributele pentru imagini, scutind astfel browserul sa faca niste calcule pentru a stabili unde amplaseaza celelalte elemente (text, tabele, etc.).

2. Pentru a folosi o imagine ca fond pentru o pagina web se foloseste atributul **background** al etichetei <BODY> astfel:

<BODY background="adresa_URL_a_imaginii">

Imaginea se va multiplica aliniat (tiling) pe orizontala si verticala pana umple ecranul.

3. Pentru a folosi imaginea ca o legatura se va folosi in locul textului cu eticheta <A> astfel:

** **

Imagini video

Pentru a incarca o imagine video se foloseste atributul dynsrc in loc de src, astfel:

Singurul format de fisiere video care este suportat de Internet Explorer este **.AVI** (Audio Video Interleave). Netscape nu le recunoaste, iar pentru a vizualiza o imagine video *on-line* in acest browser trebuie instalat un program auxiliar de tip **plug-in** (pe care ar trebui sa-l aiba si utilizatorul paginii). Datorita acestui motiv ar trebui incluse in cadrul etichetei si a unei imaginii statice ce va fi afisata daca browserele nu recinosc extensia dynsrc. Ordinea in care apar cele doua atribute nu are importanta ().

Atribute:

controls – pentru a adauga butoane de control standard, ca la aparatele video (nu are valori).

loop – pentru a repeta clipul video de un numar precizat de ori sau la infinit (pentru valoarea **infinite**).

start – pentru a preciza cand incepe derularea imaginii video (**mouseover** – adica din momentul in care mouseul este plasat deasupra imaginii video, **fileopen** – derularea imaginii incepe imediat dupa incarcarea ei in pagina, valoare implicita). Cele doua valori pot fi puse si amandoua despartite prin virgula pentru a porni imediat si apoi dupa plasarea mouseului deasupra.

Sunete

<BGSOUND> - permite includerea unei muzici de fundal. Ea este o extensie doar a browserului Internet Explorer.

<BGSOUND src="URL_fisier_sunet" loop="valoare">

Permite redarea a trei tipuri de fisiere de sunet:

- **.wav** un format nativ pentru PC
- **.au** format nativ pentru sistemele UNIX
- **.midi** format universal acceptat pentru codificarea sunetelor.

Atributul loop are valorile ca si cele pentru videoclipuri.

Liste

- **liste neordonate (Unordered List)**

Fiecare element al listei este introdus prin eticheta List Item Intrucat dupa o eticheta urmeaza alta, nu este obligatoriu, cu toate ca aceasta e o eticheta container.

La inceputul fiecarui element al fistei vor se va afisa cu cate o „bulina” bullet stabilita prin atributul **type** ce poate fi definit atat in eticheta cat si in eticheta . Valorile sale pot fi:

- ~ circle (cerc)
- ~ disc (disc plin) – valoare prestabilita
- ~ square (patrat)

Observatie: Mai exista doua forme particulare de liste neordonate:

- ~ lista de directoare introdusa prin <DIR> </DIR>
 - ~ lista de meniuri ce utilizeaza eticheta <MENU> </MENU>
- Aproape ca nu exista diferente intre acestea si eticheta

- liste ordonate (Ordered List)

Fiecare element al listei este introdus tot prin eticheta List Item diferenta fata de cele neordonate fiind ca in acest caz marcarea elementelor se face prin cifre. In acest caz, valorile atributului type din sau pot fi:

- ~ 1 – pentru ordonare de tipul 1, 2, 3 ... (cifre arabe – ordine prestabilita)
- ~ I – pentru ordonare de tipul I, II, III, IV, ... (cifre romane mari)
- ~ i – pentru ordonare de tipul i, ii, iii, iv, ... (cifre romane mici)
- ~ A – pentru ordonare de tipul A, B, C, ... (litere mari)
- ~ a – pentru ordonare de tipul a, b, c, ... (litere mici)

De asemenea, eticheta poate avea atributul **start**, care stabileste valoarea initiala a secventei de ordonare. Valoarea acestui atribut trebuie sa fie un numar intreg pozitiv.

Observatie. Listele neordonate si ordonate pot fi imbricate unele in altele fara restrictii.

- liste de definitii (Definition List) care nu au nici un fel de marcaj

<DL> </DL>

Fiecare element al listei este introdus prin eticheta <DT> (Definition Term) iar acest element contine la randul sau un numar de elemente care il definesc, introduse prin eticheta <DD> (Definition Description). Ca si in cazul etchetei , etichetele de inchidere </DT> si </DD> sunt optionale.

Tabele

<TABLE> </TABLE>

<CAPTION> Table Caption – pentru a atasa un titlu tabeli. Se plaseaza in interiorul etichetei <TABLE> dar nu in interiorul etchetelor <TR> sau <TD>.

<TR> </TR> *Table Row* - pentru a introduce un rand (o linie) in tabel

<TD> </TD> *Table Data* – pentru a introduce celulele de pe linia respectiva.

Celulele tabelului pot sa contina: text, imagini, link-uri, formulare etc. (fiecare celula avand propriile optiuni (attribute) pentru culoarea fondului, culoarea textului, alinierea textului, etc.

<TH> </TH> *Table Header* - pentru a introduce celule cu semnificatia de cap de tabel in cadrul unui rand definit cu <TR>. Toate attributele care pot fi atasate etichetei <TD> pot fi de asemenea atasate si etichetei <TH>. Continutul celulelor definite cu <TH> este scris implicit cu caractere aldine si centrat.

Atribute etchetei <TABLE> sunt:

border – are valori intregi pozitive reprezentand grosimea in pixeli a chenarului tabelului (0 pentru absenta lui)

align – pentru alinierea tabelului in pagina (left, valoare prestabilita – textul care urmeaza dupa punctul de inserare al tabelului va fi dispus in partea dreapta a tabelului, right – textul care urmeaza dupa punctul de inserare al tabelului va fi dispus in partea stanga a tabelului, center – textul care urmeaza dupa punctul de inserare al tabelului va fi afisat pe toata latimea paginii imediat sub tabel)

width – pentru a stabili exact latimea tabelului in pixeli sau procente

height - pentru a stabili exact inaltimea tabelului in pixeli sau procente

cellspacing – pentru a stabili distanta in pixeli dintre doua celule vecine ale tabelului. Valoarea prestabilita e de 2 pixeli.

cellpadding – pentru a stabili distanta in pixeli dintre marginea unei celule si continutul ei. Valoarea prestabilita e de 1 pixel.

bgcolor – pentru a stabili culoarea de fond a tabelului

background – pentru a stabili ca fond al unui tabel o imagine

bordercolor – pentru a stabili culoarea chenarului tabelului

bordercolorlight – pentru a stabili culoarea marginilor din stanga si de sus ale tabelului

bordercolordark – pentru a stabili culoarea marginilor din dreapta si de jos ale tabelului

frame – permite specificarea laturilor din chenarul unui tabel care pot fi vizibile (**void** – elimina toate muchiile exterioare ale unei tabele, **above** – afiseaza muchia in partea superioara a tabelului, **below** –

afiseaza muchia in partea inferioara a tabelului, **hsides** – afiseaza cate o muchie in partea inferioara si in cea superioara a tabelului, **vsides** – afiseaza cate o muchie in partea din stanga si in cea din dreapta a tabelului, **lhs** – afiseaza o muchie in partea din stanga a tabelului, **rhs** – afiseaza o muchie in partea din dreapta a tabelului, **box** – afiseaza o muchie pe toate laturile cadrului tabelului).

rules – permite alegerea unor delimitatori intre celulele unui tabel (**none** – elimina toate muchiile interioare ale tabelului, **rows** – afiseaza muchii interioare intre toate liniile tabelului, **cols** – afiseaza muchii verticale intre toate coloanele tabelului, **all** – afiseaza muchii intre toate liniile si coloanele tabelului).

hspace – precizeaza distanta in pixeli pe orizontala dintre tabel si restul elementelor din pagina.

vspace – precizeaza distanta in pixeli pe verticala dintre tabel si restul elementelor din pagina.

Atributele hspace si vspace sunt recunoscute numai de browserele Netscape.

Atribute etchetei <CAPTION> sunt:

Align – pentru aliniere (bottom – sub tabel, top – deasupra tabelului, left – deasupra la stanga tabelului, right – deasupra, la dreapta tabelului).

Atribute etchetei <TR> sunt:

bgcolor – pentru a stabili culoarea de fond a randului.

background – pentru a stabili ca fond al unui rand o imagine

Atribute etchetei <TD> sau <TH> sunt:

width – pentru a stabili exact latimea celulei in pixeli sau procente din latimea tabelului

height - pentru a stabili exact inaltimea celulei in pixeli sau procente din inaltimea tabelului

align – pentru alinierea continutului unei celule (left, right, center)

valign – pentru alinierea continutului unei celule pe verticala (baseline – la baza, bottom – jos, middle – la mijloc (valoarea prestabilita), top – sus)

bgcolor – pentru a stabili culoarea de fond a celulei

background – pentru a stabili ca fond al unei celule o imagine

colspan – realizeaza extinderea unei celule peste celulele din dreapta ei (valoarea atributului specifica numarul de celule care se unifica).

rowspan – realizeaza extinderea unei celule peste celulele de sub ea (valoarea atributului specifica numarul de celule care se unifica).

Sunt posibile extinderi ale unei celule atat pe orizontala cat si pe verticala (in acest caz vor fi prezente ambele atribute: colspan si rowspan).

Observatii.

1. Daca intr-un tabel sunt definite mai multe atribute bgcolor, atunci culoarea de fond cu prioritatea cea mai mica o are <TABLE>, apoi <TR>, iar cea mai mare prioritate o are cea stabilita in cadrul etichetei <TD>.

2. Culoarea textului din fiecare celula se poate stabili ajutorul etichetei

3. Daca intr-un tabel exista celule fara continut (vide), atunci aceste celule vor aparea in tabel fara un chenar de delimitare. Pentru a afisa un chenar in jurul celulelor vide se poate adauga dupa <TD> fie fie
.

4. Un tabel poate sa contina in celulele sale si alte elemente, in afara de text, inclusiv un alt tabel, formand astfel un ansamblu de tabele imbricate.

5. Eticheta <COLGROUP> </COLGROUP> permite definirea unui grup de coloane carora li se pot atribui diverse elemente de formatare in cadrul unui tabel.

Atributele acceptate de <COLGROUP> sunt:

span – determina numarul de coloane dintr-un grup.

width – determina o latime unica pentru coloanele din grup.

align – determina un tip unic de aliniere pentru coloanele din grup.

Intr-un bloc <COLGROUP>, coloanele pot avea configurari diferite daca se utilizeaza eticheta <COL>, care admite atributele:

span – identifica acea coloana din grup pentru care se face configurarea. Daca lipseste, atunci coloanele sunt configurate in ordine.

width – determina latimea coloanei identificate prin span.

align – determina alinierea continutului coloanei identificate prin span.

Cadre (frames)

Este posibilă împărțirea ferestrei browserului în mai multe ferestre distincte, denumite **cadre (frames)** putând astfel să afișăm în aceeași fereastră a browserului mai multe documente HTML diferite, câte unul în fiecare cadru.

Pentru aceasta avem nevoie de:

- un document de definire a cadrelor (în care sunt etichete ce stabilesc numărul, dimensiunile și așezarea documentelor în pagină);

- câte un fișier HTML pentru fiecare cadru în parte, prin care se stabilește conținutul cadrului respectiv.

În documentul de definire a cadrelor, blocul `<BODY> </BODY>` este înlocuit de blocul `<FRAMESET>`

`</FRAMESET>` iar în interiorul acestuia, fiecare cadru este introdus prin eticheta `<FRAME>`.

`<FRAMESET attribute> </FRAMESET>`

Atribute FRAMESET:

cols – stabilește în câte cadre de tip coloană se împarte fereastra browserului (valoarea este o listă de elemente separate prin virgulă ce descrie modul în care se face împărțirea – dacă e un număr în întreg reprezintă câți pixeli, dacă e un număr cuprins între 1-99 urmat de % înseamnă că e un procent din fereastra browserului, iar valorile de tipul n^* semnifică faptul că coloana astfel definită ocupă a n-a parte din spațiul rămas după dispunerea în fereastra browserului a coloanelor precedente)

rows – stabilește în câte cadre de tip linie se împarte fereastra browserului (valoarea este o listă de elemente în mod similar ca la coloane).

bordercolor – stabilește culoarea tuturor marginilor cadrelor incluse.

border – stabilește lățimea tuturor chenarelor cadrelor incluse (valoarea e un număr ce reprezintă grosimea în pixeli – valoarea prestabilită fiind de 5 pixeli, valoarea 0 reprezintă cadre fără chenar).

frameborder – activează/dezactivează afișarea marginii tuturor cadrelor (valoarea „yes” o activează iar valoarea „no” o dezactivează, în acest caz e echivalent cu `border="0"`).

`<FRAME src="adresa_URL_document_HTML"> </FRAME>`

Atribute FRAME:

src (source) – conține adresa URL a documentului ce se va încărca în acel cadru.

bordercolor – stabilește culoarea marginilor cadrului definit.

frameborder – activează/dezactivează afișarea marginii cadrului definit (valoarea „yes” o activează iar valoarea „no” o dezactivează, în acest caz e echivalent cu `border="0"`).

noresize – are ca efect interzicerea redimensionării cadrului cu ajutorul mouseului de către utilizator.

scrolling – adaugă cadrului respectiv o bară de derulare sau de defilare (scrolling bar) care permite navigarea în interiorul documentului afișat în cadru (valorile posibile sunt: **yes** – bară de derulare este prezentă totdeauna, **no** – bară de derulare nu este disponibilă și **auto** – bară de derulare este vizibilă atunci când e necesar, în funcție de dimensiunile textului din cadrul respectiv).

marginheight – stabilește distanța în pixeli dintre conținutul unui cadru și marginile verticale ale cadrului (un număr de pixeli sau un procent din înălțimea cadrului)

marginwidth – stabilește distanța în pixeli dintre conținutul unui cadru și marginile orizontale ale cadrului (un număr de pixeli sau un procent din lățimea cadrului).

Cadre interne (in-line frames)

Prin intermediul lor se poate împărți fereastra browserului în mai multe zone dreptunghiulare (numite cadre sau frame) în care se pot încărca documente HTML diferite. Pentru a utiliza cadre, trebuie creat un document special (denumit document FRAMESET), care să definească împărțirea în cadre a ferestrei browserului, dimensiunea, poziția și rolul fiecărui cadru. Sintaxa acestui document este:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <FRAMESET>
    ...
  </FRAMESET>
</HTML>
```

Atributele elementului **FRAMESET** sunt:

ROWS=listă_dimensiuni

COLS=listă_dimensiuni

unde **listă_dimensiuni** reprezintă dimensiunea cadrelor dată în pixeli, procente sau utilizând * cu semnificația „ocupă întreg spațiul rămas”. De exemplu:

```
<FRAMESET ROWS="50%,50%" COLS="50*,100"> ... </FRAMESET>
```

defineste o rețea de două cadre orizontale, împărțite fiecare în câte trei cadre verticale (total 6 cadre).

Cadrele pot fi imbricate (adică pot fi incluse în interiorul altor cadre). De exemplu:

```
<FRAMESET COLS="100,*" >...<FRAMESET ROWS="50%,50%"> ... </FRAMESET> </FRAMESET>
```

Împarte ecranul în trei cadre: unul vertical la stanga de 100 pixeli, iar la dreapta cel de-al doilea vertical (care ocupă restul spațiului disponibil) și este împărțit în două cadre orizontale de dimensiuni egale.

Există browsere care nu suportă cadre. Pentru acestea se utilizează în interiorul blocului <FRAMESET> eticheta <NOFRAMES> </NOFRAMES> între care se pot introduce orice alte etichete HTML (inclusiv imagini, hiperlink-uri, tabele). Dacă browserul poate să interpreteze cadre, va ignora ce se găsește în porțiunea dintre <NOFRAME> și </NOFRAME>, iar dacă browserul nu poate interpreta cadre, materialul cuprins între <NOFRAME> și </NOFRAME> va fi singurul recunoscut și afișat.

Prin intermediul elementului **<FRAME attribute>** se specifică pentru fiecare cadru în parte conținutul și aspectul. Atributele care fac acest lucru sunt:

NAME=șir_de_caractere ce asociază un nume cadrului respectiv ce poate fi utilizat ca țintă (target) pentru legături din alte cadre;

SRC=URL specifică adresa documentului HTML ce va fi deschis inițial în cadru;

LONGDESC=URL specifică adresa la care se găsește o descriere detaliată a acestui cadru (utilă pentru browser-ele care nu pot vizualiza cadre);

NORESIZE specifică faptul că dimensiunea cadrului respectiv nu poate fi modificată trăgând cu mouse-ul (drag-and-drop), în mod implicit putând-o modifica;

SCROLLING=AUTO | YES | NO specifică modul de defilarea informațiilor în cadru (**AUTO** – barele de defilare sunt vizualizate doar dacă e necesar (valoare implicită), **YES** - barele sunt permanent vizibile, **NO** – nu sunt vizualizate bare de defilare);

FRAMEBORDER=1 | 0 specifică dacă frame-ul este sau nu seste separat de cadrele adiacente prin linii de contur (valoarea implicită este 1);

MARGINWIDTH=număr_de_pixeli și **MARGINHEIGHT=număr_de_pixeli** specifică spațiul dintre conținutul cadrului și marginile stânga și dreapta, respectiv sus și jos (valoare minimă fiind 1, iar cea implicită depinde de fiecare browser);

Exemplu.

....

Furmulare (FORM)

<FORM> elemente_control_formular </FORM>

elemente_control_formular pot fi:

- **Câmpuri text** ce permit utilizatorului să introducă un text format dintr-o singură linie (prin intermediul elementului **INPUT**) sau mai multe linii (prin intermediul elementului **TEXTAREA**);
- **Butoane de validare (checkbox)** ce permit utilizatorului să selecteze sau nu anumite opțiuni prin activarea sau dezactivarea butoanelor asociate. Este o selecție fără excludere (utilizatorul putând selecta zero, unul sau mai multe dintre opțiuni). Aceste butoane pot fi incluse în formular prin intermediul elementului **INPUT**.
- **Butoane radio** ce sunt de tip excludere (utilizatorul poate selecta doar unul, celelalte fiind automat dezactivate). Aceste butoane pot fi incluse în formular prin intermediul elementului **INPUT**.
- **Butoane de comandă** (de trei tipuri: **Submit** – când sunt activate, transmit informațiile înscrise în formular către server, **Reset** – resetează formularul (toate câmpurile formularului sunt completate cu valorile lor inițiale), **Push** – execută o acțiune definită de autorul formularului)
- **Meniuri** introduse prin elementul **SELECT** în combinație cu **OPTION** și **OPTGROUP**
- **Elemente de control ascunse (hidden)** ce permit autorului formularului ca odată cu informațiile înscrise de utilizator în formular să transmită și alte informații utile și se înserează în formular cu ajutorul elementului **INPUT**.

Elementul <INPUT attribute> unde **attribute** poate fi:

NAME=șir_de_caractere ce asociază elementului de control un nume

VALUE=șir_de_caractere ce specifică valoarea inițială a elementului de control (opțional, cu excepția cazului când tipul specificat este **RADIO**)

TYPE=TEXT | PASSWORD | CHECKBOX | RADIO | SUBMIT | RESET | BUTTON | FILE | HIDDEN | IMAGE ce specifică tipul elementului de control creat (valoarea implicită este **TEXT**).

Tipul **PASSWORD** permite inserarea în formular a unui câmp de introducere a parolei (asemănător cu câmpul **TEXT**).

Tipurile **SUBMIT**, **RESET** și **BUTTON** permit inserarea butoanelor de comandă **Submit**, **Reset** și **Push**.

Tipul **IMAGE** este utilizat atunci când dorim să specificăm o altă imagine pentru butonul **SUBMIT**.

Tipul **FILE** permite utilizatorului să specific un fișier ce va fi transmis prin intermediul formularului.

SIZE=valoare ce specifică dimensiunea (în pixeli, cu excepția tipurilor **TEXT**, **FILE** sau **PASSWORD** când se specifică în număr de caractere) inițial a elementului de control.

MAXLENGTH=valoare se utilizează în cazul tipurilor **TEXT** sau **PASSWORD** și specifică numărul maxim de caractere pe care le poate introduce utilizatorul.

Elementul **<BUTTON attribute> ... </BUTTON>** care, spre deosebire de elementul **INPUT**, admite etichetă de început și etichetă de sfârșit. Conținutul elementului **BUTTON** va fi vizualizat pe butonul creat (poate fi și o imagine). Acest element admite ca atribute principale: **NAME**, **VALUE** și **TYPE** (cu valorile posibile: **SUBMIT**, **RESET**, **BUTTON**).

Elementul **<SELECT attribute> ... </SELECT>** permite introducerea de meniuri într-un formular. Atributele sale sunt:

NAME=șir_de_caractere ce asociază elementului de control un nume

SIZE=valoare ce specifică numărul de opțiuni vizibile în fereastra derulantă în care e vizualizat meniul.

MULTIPLE permite selectarea mai multor opțiuni din meniu (introduse cu elementul **OPTION**).

Elementul **<OPTION attribute> ... </OPTION>** permite introducerea opțiunilor din care este constituit meniul. Atributele sale sunt:

SELECTED ce specifică faptul că această opțiune este selectată inițial.

VALUE=șir_de_caractere ce specifică valoarea inițială a elementului de control (dacă nu e specificat, se consideră valoarea inițială a elementului **OPTION**).

LABEL=șir_de_caractere ce specifică un nume mai scurt (etichetă) pentru această opțiune.

Elementul **<TEXTAREA attribute> ... </TEXTAREA>** permite inserarea unui câmp text din mai multe linii. Atributele sale sunt: pe linie.

ROWS=număr_de_linii specifică numărul de linii vizibile din textul respective.

COLS=număr_de_caractere specifică numărul maxim de caractere

Utilizatorul navighează printre elementele de control ale formularului cu ajutorul tastei **TAB**. Pentru a stabili ordinea în care vor fi parcurse acestea, se folosește atributul:

TABINDEX=valoare unde **valoare** e un număr cuprins între 0 și 32767. Dacă sunt mai multe cu același număr de ordine, ele vor fi parcurse în ordinea în care sunt descrise.

Asocierea unei taste de control pentru un element de control din formular se face cu atributul următor:

ACCESSKEY=character și ca urmare elemental respective va fi accesat rapid acționând tasta **ALT** și **character**

De menționat că pe lângă aceste elemente de control, în formular mai pot fi inserate: texte și elemente de formatare a textelor respective, grafică, link-uri, etc.

Transmiterea datelor introduse într-un formular

La acționarea butonului **SUBMIT**, toate informațiile introduse de utilizator sunt transmise către server sub forma unei perechi (**nume_câmp**, **valoare**) și vor fi prelucrate de către un program special denumit **program** (sau **script**) **CGI** (**Common Gateway Interface**). Specificarea acestui program se realizează prin intermediul atributului următor:

ACTION=URL unde **URL** specifică adresa programului **CGI**.

Modalitatea (metoda protocolului **HTTP**) în care sunt transmise informațiile către server este stabilită de atributul următor:

METHOD= POST | GET în cazul valorii **POST**, informațiile sub formă de perechi (**nume_câmp**, **valoare**) sunt incluse în corpul formularului și transmise către programul **CGI** care va realiza prelucrarea lor, iar în cazul valorii **GET** informațiile introduse de utilizator sunt atașate adresei **URL** care specifică programul ce va realiza prelucrarea lor, separate prin **?**, apoi acest nou **URL** este transmis către programul **CGI** de prelucrare a informațiilor.

Stiluri CSS (*Cascade Style Sheets* – șabloane de formatare text construite

în cascadă)

CSS sau stilurile de formatare, au fost elaborate începând cu 1994, devenind o recomandare a W3C¹ (World Wide Web Consortium) în 1996. Ele constau în stabilirea de la început a tuturor formatărilor necesare pentru documentele HTML (cum ar fi: fonturi, culori, margini, spațierea liniilor, etc.). Avantaje:

- *Economie de timp.* De exemplu, dacă dorim să colorăm în albastru toate paragrafele titlu de nivel 1 (Heading 1) putem să aplicăm atributului H1 un stil în mod unitar, fără a specifica de fiecare dată formatul dorit.
- *Stilurile sunt ușor de modificat.* De exemplu, pentru a schimba culoarea paragrafelor titlu de nivel 1 din albastru în verde, nu este necesar să parcurgem întregul document HTML, înlocuind culoarea, ci este suficient să modificăm descrierea stilului elementului H1.
- *Stilurile sunt aplicate de browser în mod unitar.* Documentele HTML fiind extrem de stufoase, este foarte ușor să greșim. Putem uita un atribut, putem să uităm eticheta de sfârșit a unui element etc. Dar stilurile de formatare vor fi aplicate în mod unitar, în întregul document HTML.
- *Stilurile oferă posibilități de formatare suplimentare, care nu sunt incluse în documentul HTML.* De exemplu, putem specifica spațierea liniilor sau a caracterelor, putem configura marginile sau putem utiliza chenare.
- *Stilurile pot fi aplicate mai multor pagini Web.* Astfel, putem crea mai multe pagini Web cu aspect unitar. Orice modificare al stilurilor va fi automat reflectată în toate paginile Web care le utilizează. În acest caz, stilurile de formatare trebuie să fie memorate în fișiere separate.

Există și alte limbaje de definire a stilurilor, dar cel mai utilizat este CSS. Pentru a specifica limbajul pentru definirea stilurilor utilizat va fi inclusă în antetul documentului HTML o declarație prin intermediul elementului **META** astfel:

<META HTTP-EQUIV = "Content-Style-Type" CONTENT="text/css">

Precizarea stilurilor de formatare care se aplică într-un document HTML se poate face în trei moduri: *inline*, în *antet* sau *extern*.

1. Specificarea stilurilor *inline* se realizează prin intermediul atributului **STYLE**:
STYLE=șir de caractere unde **șir de caractere** este un mod de a descrie formatarea ce e specifică fiecărui limbaj de descriere a stilurilor. De exemplu, pentru a specifica faptul că paragraful curent este aliniat centrat și scris cu roșu, putem scrie folosind CSS astfel:
<P STYLE="TEXT-ALIGN: CENTER; COLOR: RED"> Am aplicat un stil
Specificarea *inline* este utilă atunci când stilul respectiv se aplică unui singur element.
2. Specificarea stilurilor *în antet* se realizează prin intermediul elementului **STYLE**. Acest element ne permite să includem în antetul documentului HTML toate declarațiile de stiluri utilizate în cadrul lui (atributul **TYPE** este obligatoriu). De exemplu:
<STYLE TYPE="text/css">
P {FONT: 12 pt Arial; FONT-STYLE: Italic; TEXT-ALIGN: Left}
</STYLE>
Putem utiliza și clase (vom vedea la tipul extern ce sunt acestea). Se pot defini și stiluri care să facă excepție de la stilul curent, cu ajutorul unui identificator, astfel:
<STYLE TYPE="text/css">
#Excepție {FONT-SIZE:10 pt; TEXT-ALIGN:Right}
</STYLE>
iar în document precizăm că utilizăm acest stil astfel:
<P ID=Excepție> Acest paragraph este o excepție de la stilul current </P>
3. Specificarea stilurilor în mod *extern* este cel mai flexibil mod de lucru cu stiluri de formatare, deoarece se pot utiliza aceleași stiluri pentru mai multe documente HTML, iar modificarea unui stil de formatare nu presupune modificarea documentului HTML. Pentru a specifica fișierele în care sunt definite stilurile, se utilizează în antetul documentului HTML elementul **LINK**:
<LINK attribute>
Prin intermediul atributului **TYPE=șir de caractere** se specifică limbajul de descriere, atributul **HREF=șir de caractere** specifică adresa fișierului care conține descrierea stilurilor, iar atributul **REL=șir de caractere** dacă are valoarea **"alternate stylesheet"**, atunci browser-ul oferă

¹ Specificația originală elaborată de World Wide Web Consortium se poate consulta la adresa <http://www.w3.org/pub/WWW/TR/REC-CSS1>.

vizitatorului paginii repective posibilitatea de a alege una dintre variante. Dacă dorim să precizăm care dintre variantele de stiluri posibile este preferat, îi asociem un nume (prin intermediul elementului **TITLE**) și asociem atributului **REL** valoarea „**stylesheet**” ca în exemplul următor:
<LINK TYPE="text/css" REL="stylesheet" HREF="unstil.css" TITLE="De preferat">

Definiția unui stil conține un element HTML (denumit selector) și una sau mai multe declarații care stabilesc modul de vizualizare a elementului respectiv. Declarațiile sunt incluse între paranteze acolade și sunt separate prin caracterul “;”. O declarație este constituită din două părți: o proprietate și valoarea atribuită proprietății respective. Proprietatea este separată de valoare prin caracterul “:”.

selector {proprietate:valoare}

Exemplu: **P {color:red}** unde selectorul este elementul HTML **P**, proprietatea este **color**, iar valoarea atribuită este **red**. Ca efect, toate paragrafele vor fi scrise cu roșu.

Selectorul are rolul de a preciza asupra cărui element HTML se aplică declarațiile de formatare.

Dacă nu dorim ca toate aparițiile unui element HTML să fie formate în același fel, putem constitui clase.

Pentru a declara faptul că un element HTML aparține unei anumite clase, în documentul HTML vom utiliza atributul **CLASS** în descrierea elementului respectiv, astfel: **CLASS=șir de caractere**

Pentru a defini un stil pentru o anumită clasă a unui element HTML, utilizăm drept selector atât numele elementului cât și numele clasei, separate prin caracterul “.”. De exemplu, putem defini stilul următor:

P.observatie {FONT-STYLE=italic; BORDER-WIDTH=0.5; BORDER-STYLE=DOUBLE}

iar în documentul HTML pentru a preciza că pentru paragraph se aplică acest stil, se procedează astfel:

<P CLASS="observatie"> Aici e scris un text

În cazul declarării unui stil **inline** (prin intermediul atributului **STYLE**), declarațiile nu sunt încadrate între paranteze acolade, ci între ghilimele.

Proprietățile prin intermediul cărora putem controla formatarea textelor utilizând CSS, sunt: **COLOR, BACKGROUND, FONT-FAMILY, FONT-STYLE, FONT-WEIGHT, FONT-SIZE, TEXT-DECORATION, TEXT-TURN, LINE-HEIGHT, WORD-SPACING, LETTER-SPACING, TEXT-INDENT, WHITE-SPACE, TEXT-ALIGN, A, BORDER, MARGIN, PADDING, LIST-STYLE-TYPE.**

COLOR – precizează culoarea textului

BACKGROUND – precizează fundalul utilizat

FONT-FAMILY – permite stabilirea fontului utilizat pentru text

FONT-STYLE – permite aplicarea diferitelor stiluri pentru font. Valori posibile: **italic/oblique** (pentru text scris înclinat) și **normal** (pentru text scris normal).

FONT-WEIGHT – stabilește grosimea textului. Valori posibile: un număr din mulțimea **{100,200,300,400,500,600,700}**, **bold** (grosime medie, echivalentul valorii 700), **bolder** (grosime mai mare), **lighter** (grosime mai mică), **normal** (text normal, echivalentul valorii 400).

FONT-SIZE – stabilește dimensiunea caracterelor. Putem specifica dimensiunea caracterelor exact în număr de puncte tipografice (**pt**) sau în număr de pixeli (**px**) sau putem utiliza un descriptor absolut: **xx-small, x-small, small, medium, large, x-large, xx-large**. Dacă dorim să specificăm dimensiunea caracterelor relativ la dimensiunea inițială, putem utiliza descriptorii **bigger** sau **smaller** sau să precizăm dimensiunea în procente.

TEXT-DECORATION – permite efecte de evidențiere a textului. Valori posibile: **underline** (textul e subliniat), **overline** (textul e supraliniat), **line-through** (textul e tăiat cu o linie), **blink** (textul clipește), **none** (nu se utilizează nici un efect).

TEXT-TURN – permite controlul tipului de litere utilizate în text. Valori posibile: **uppercase** (toate literele vor fi litere mari), **lowercase** (toate literele vor fi litere mici), **capitalize** (prima literă din cuvânt va fi majusculă), **none** (textul rămâne așa cum a fost scris).

LINE-HEIGHT – stabilește distanța dintre liniile unui paragraf. Spațierea liniilor se poate specifica în număr de puncte tipografice (**pt**) sau în număr de pixeli (**px**) sau procentual relativ la dimensiunea curentă

a caracterelor.

WORD-SPACING – permite stabilirea distanței dintre două cuvinte consecutive. Distanța se poate specifica în număr de puncte tipografice (**pt**) sau în număr de pixeli (**px**) sau în număr de **em** (un **em** este distanța necesară pentru a scrie litera **M**; prin urmare această mărime depinde de fontul folosit).

LETTER-SPACING - permite stabilirea distanței dintre literele unui cuvânt exprimată în număr de puncte tipografice (**pt**) sau în număr de pixeli (**px**) sau în număr de **em**.

TEXT-INDENT – stabilește distanța dintre marginea din stânga a textului și marginea zonei de vizualizare. Distanța se specifică în număr de puncte tipografice (**pt**) sau în număr de pixeli (**px**) sau procentual. Valorile pot fi atât pozitive (în acest caz textul este deplasat la dreapta față de marginea zonei de vizualizare), cât și negative (textul este deplasat la stânga).

WHITE-SPACE – permite controlul spațiilor dintre cuvinte. Valori posibile: **pre** (browser-ul va afișa și spațiile suplimentare dintre cuvinte și liniile albe, asemănător cu elementul HTML **PRE**), **nowrap** (textul unui paragraf este scris tot pe o singură linie, exceptând cazul în care forțăm trecerea la linie nouă prin intermediul elementului HTML **BR**)

TEXT-ALIGN – permite stabilirea modului de aliniere a textului față de marginile din stânga-dreapta ale zonei de vizualizare. Valori posibile: **left** (textul este aliniat la marginea din stânga), **right** (textul este aliniat la marginea din dreapta), **center** (textul este aliniat la centru), **justify** (textul este aliniat la ambele margini).

A – permite stabilirea aspectului legăturilor.

A: link {proprietate: valoare} definește aspectul legăturilor nevizitate;

A: visited {proprietate: valoare} definește aspectul legăturilor vizitate;

A: active {proprietate: valoare} definește aspectul legăturii active;

A: hover {proprietate: valoare} definește aspectul legăturii asupra căreia este plasat cursorul mouse-ului;

A: {proprietate: valoare} modifică aspectul tuturor acestor legături.

BORDER – permite trasarea unui chenar în jurul unui element HTML. Caracteristicile chenarului sunt stabilite prin intermediul proprietăților **BORDER-COLOR** (culoarea), **BORDER-WIDTH** (grosimea liniei chenarului, specificată în număr de pixeli sau prin intermediul descriptorilor **thin**, **medium**, **thick**), **BORDER-STYLE** (tipul de linie, specificată prin intermediul descriptorilor **solid**, **double**, **ridge**, **dashed**, **dotted**, **groove**, **inset**, **outset**, **none**).

MARGIN – stabilește distanța (în pixeli) dintre text și marginile zonei de vizualizare:

MARGIN-LEFT – distanța față de marginea din stânga;

MARGIN-TOP – distanța față de marginea de sus;

MARGIN-BOTTOM – distanța față de marginea de jos;

MARGIN-RIGHT – distanța față de marginea din dreapta.

PADDING – stabilește distanța (în puncte tipografice) dintre chenarul unui element și conținutul său (text sau imagine):

PADDING -LEFT – distanța față de marginea din stânga;

PADDING -TOP – distanța față de marginea de sus;

PADDING -BOTTOM – distanța față de marginea de jos;

PADDING -RIGHT – distanța față de marginea din dreapta;

PADDING – toate cele patru distanțe.

LIST-STYLE-TYPE – descrie modul de marcare a intrărilor într-o listă. Valori posibile: **disc** (valoare implicită), **circle**, **square**, **decimal**, **lower-roman**, **upper-roman**, **lower-alpha**, **upper-alpha**, **none**.

Observații:

1. Modul de vizualizare a stilurilor depinde de browser-ul utilizat.
2. Majoritatea proprietăților acceptă valoarea none (sau normal), ceea ce pare a nu avea nici un efect. Ea este însă foarte utilă atunci când dorim să contracărăm efectul unor alte stiluri de formatare.
3. Pentru a descrie mai concis un stil de formatare, putem grupa proprietățile de formatare. De exemplu, pentru text putem folosi denumirea comună FONT, apoi le putem specifica în următoarea ordine: **FONT-STYLE, FONT-WEIGHT, FONT-SIZE, LINE-HEIGHT, FONT-FAMILY**. Nu este necesar să le specificăm pe toate (în rest, se vor utiliza valorile implicite). De exemplu, putem să scriem:
H1 {FONT: italic, bold, 14pt/120% Arial}
 cu observația că valoarea proprietății **LINE-HEIGHT** este separată prin caracterul / (slash) de valoarea proprietății **FONT-SIZE**.
4. La fel putem să grupăm cele trei elemente de formatare a chenarelor (**BORDER-COLOR, BORDER-WIDTH, BORDER-STYLE**) putând fi reunite sub denumirea comună **BORDER**.
5. Într-un document HTML putem utiliza atât stiluri definite **extern**, stiluri definite **intern (în antet)**, cât și stiluri definite local (**inline**). Prin urmare, documentul HTML acceptă „o **cascadă**” de stiluri. Este posibil ca unele proprietăți să fie definite de mai multe ori sau deloc. În această situație, ultima definiție este prioritară.
 Dacă o proprietate a unui element nu este definită deloc, atunci browser-ul analizează dacă ea nu poate fi moștenită. De exemplu, dacă am stabilit un stil pentru elementul **BODY**, în care precizăm fontul utilizat, iar pentru elementul **P** nu am specificat fontul, elementul **P** va moșteni fontul definit în stilul elementului **BODY**. În cazul în care proprietatea respectivă nu poate fi moștenită, browser-ul utilizează valorile sale implicite.

Scripturi JAVASCRIPT

Un **script** este un program ce poate fi inclus într-un document HTML sau poate fi asociat unui astfel de document și care este executat pe calculul utilizatorului, fie la încărcarea documentului HTML, fie la producerea unui anumit eveniment (ca de exemplu: un clic, poziționarea cursorului mouse-ului într-o anumită zonă, mișcarea mouse-ului).

Scripturile pot fi de două tipuri:

- Care se execută o singură dată, când este încărcat documentul HTML (acestea sunt incluse în documentul HTML prin intermediul elementului <SCRIPT>);
- Care se execută ori de câte ori se produce un eveniment (acestea pot fi asociate elementelor HTML prin intermediul unor atribute specifice).

Limbajul de descriere a scripturilor este specificat în două moduri:

- 1) În antet prin intermediul elementului <META>:

<META HTTP-EQUIV="Content-Script-Type" CONTENT="tip_limbaj_script">

unde **tip_limbaj_script** este tipul limbajului de descriere a scripturilor folosit (de exemplu, tipul poate fi:

“text/tcl”, “text/javascript”, “text/vbscript”, etc.). }n cazul nostrum, pentru limbajul JavaScript, declarația va fi

<META HTTP-EQUIV="Content-Script-Type" CONTENT="javascript">

- 2) Specificarea locală a limbajului de descriere a script-urilor se realizează pentru fiecare element SCRIPT din documentul HTML, prin intermediul atributului TYPE sau a atributului LANGUAGE.

De exemplu:

<SCRIPT TYPE="text/javascript"> ... </SCRIPT>

sau

<SCRIPT LANGUAGE="Javascript"> ... </SCRIPT>

Limbajul specificat local are prioritate față de limbajul specificat în antet.

JavaScript este un limbaj de descriere a scripturilor (scripting language) bazat pe obiecte. Deși **JavaScript** și **Java** au elemente comune, nu reprezintă același lucru; **JavaScript** a fost elaborat de firma **Netscape Communications Corporation**, pe când **Java** a fost elaborat de **Sun Microsystems**.

Ambele sunt limbaje de programare orientate obiect, cu o sintaxă asemănătoare cu a limbajului **C++**.

Practic, cu ajutorul JavaScript se poate scrie și executa orice program, dar el se utilizează în principal pentru a controla modul de vizualizare a paginilor Web prin intermediul browser-ului. Principalele aplicații ale limbajului JavaScript ar fi:

- Determinarea aspectului și conținutului paginilor Web;
- Determinarea modului de funcționare a browser-ului (de exemplu, se poate modifica bara de stare

- a ferestrei browser-ului, poate deschide sau închide o fereastră, etc.);
- Prelucrarea conținutului paginii Web (de exemplu, poate valida datele introduce de utilizator prin intermediul unui formular HTML);
- Interacționează cu utilizatorul prin intermediul scripturilor asociate anumitor evenimente (de exemplu, la mișcarea mouse-ului sau executarea unui clic pe un link sau un buton, etc.);
- Interacționează cu **applet**-urile Java;
- Permite utilizarea informațiilor **cookie**.

Din motive de securitate, limbajul JavaScript nu permite citirea sau scrierea fișierelor de pe calculatorul utilizatorului.

Elementul **<SCRIPT attribute> ... </SCRIPT>** are următoarele atribute:

TYPE=șir_de_caractere specifică limbajul de descriere a script-ului

SRC=URL unde **URL** reprezintă adresa fișierului care conține script-ul (acest parametru se utilizează numai dacă script-ul este extern, memorat într-un fișier cu extensia .js). Dacă acest atribut este prezent, conținutul elementului **SCRIPT** este ignorat.

DEFER precizează browser-ului că script-ul nu va scrie nimic în documentul HTML, deci browser-ul poate continua vizualizarea documentului.

Elementul **<SCRIPT>** poate să apară de ori de câte ori, atât în antet, cât și în corpul documentului.

Limbajul JavaScript permite ca la începutul unui script să apară "**<!-**", dar ignoră celelalte caractere până la sfârșitul liniei. Deoarece JavaScript nu acceptă marcajul de sfârșit de comentariu **HTML** ("**->**") acesta va fi precedat de **//**, care are rolul de a transforma în comentariu JavaScript toate caracterele până la sfârșitul liniei.

Elementul **<NOSCRIPT> ... </NOSCRIPT>** se utilizează pentru cazul în care browser-ul utilizat nu admite script-uri sau nu suportă limbajul de descriere a script-urilor folosit.

Script-uri care se execută la apariția unui eveniment

Elementele limbajului HTML admit atribute care permit specificarea script-ului care se execută la apariția evenimentului corespunzător atributului. Aceste atribute sunt:

ONLOAD=script evenimentul producându-se când browser-ul a terminat de încărcat documentul în fereastra sa sau a terminat de încărcat toate documentele din toate cadrele definite în fereastră. Acest atribut se utilizează doar cu elementele **BODY** și **FRAMESET**.

ONUNLOAD=script evenimentul producându-se când browser-ul elimină un document din fereastră sau dintr-un cadru. Acest atribut se utilizează doar cu elementele **BODY** și **FRAMESET**.

ONCLICK=script evenimentul producându-se când se execută un clic pe element HTML corespunzător atributului (**butoane** și **link-uri**).

ONDBLCLICK=script evenimentul producându-se când se execută un dublu clic pe element HTML corespunzător atributului.

ONMOUSEDOWN=script evenimentul producându-se când se acționează butonul mouse-ului pe element HTML corespunzător atributului (un **document**, un **buton** sau un **link**).

ONMOUSEUP=script evenimentul producându-se când se eliberează butonul mouse-ului acționat pe element HTML corespunzător atributului (un **document**, un **buton** sau un **link**).

ONMOUSEOVER=script evenimentul producându-se când se deplasează cursorul mouse-ului pe link-ul HTML corespunzător atributului.

ONMOUSEMOVE=script evenimentul producându-se când se deplasează cursorul mouse-ului deasupra elementului HTML corespunzător atributului.

ONMOUSEOUT=script evenimentul producându-se când se deplasează cursorul mouse-ului de pe element HTML corespunzător atributului (un **link** sau o **image mapată**).

ONFOCUS=script evenimentul producându-se când un element de control al unui formular HTML devine active (cu ajutorul mouse-ului sau a tastei TAB). Atributul se poate utiliza doar pentru elementele **LABEL**, **INPUT**, **SELECT**, **TEXTAREA** și **BUTTON**.

ONBLUR=script evenimentul producându-se când un element de control al unui formular HTML devine inactive. Atributul se poate utiliza doar pentru elementele **LABEL**, **INPUT**, **SELECT**, **TEXTAREA** și **BUTTON**.

ONKEYPRESS=script evenimentul producându-se când se apasă și se eliberează o tastă când suntem plasați pe element HTML corespunzător atributului (un **document**, o **image**, un **link**, un **camp text**).

ONKEYDOWN=script evenimentul producându-se când se apasă o tastă când suntem plasați pe element HTML corespunzător atributului (un **document**, o **image**, un **link**, un **camp text**).

ONKEYUP=script evenimentul producându-se când se eliberează o tastă când suntem plasați pe

elemental HTML corespunzător atributului (un **document**, o **image**, un **link**, un **camp text**).

ONSUBMIT=script evenimentul producându-se când se acționează butonul SUBMIT al unui formular HTML. Se aplică numai elementului **FORM**.

ONRESET=script evenimentul producându-se când se acționează butonul RESET al unui formular HTML. Se aplică numai elementului **FORM**.

ONSELECT=script evenimentul producându-se când se selectează un text dintr-un camp text al unui formular HTML. Se aplică numai elementelor **INPUT** și **TEXTAREA**.

ONCHANGE=script evenimentul producându-se când un element de control dintr-un formular HTML și-a modificat valoarea. Se aplică numai elementelor **INPUT**, **SELECT** și **TEXTAREA**.

Toate aceste atribute primesc ca valoare un script (una sau mai multe instrucțiuni separate prin caracterul ;) încadrat între ghilimele. Script-ul se execută de fiecare dată când se produce evenimentul corespunzător atributului.

DYNAMIC HTML

Denumirea DHTML (Dynamic HTML) cuprinde trei mecanisme:

- Stiluri de formatare (style sheets);
- Mecanisme de control avansat al poziționării elementelor în pagină;
- Fonturi care pot fi descărcate pe calculatorul utilizatorului (downloadable fonts).

Stilurile de formatare și modul de descriere a acestora se realizează utilizând CSS și JavaScript.

Controlul avansat al poziționării elementelor în pagină se realizează cu ajutorul unor structuri numite **layer** ("strat", „înveliș”) ce pot fi fie transparente, fie opace. Astfel, un document HTML nu mai are o structură plană, ci este construit din mai multe straturi, care eventual se pot suprapune și a căror poziție poate fi controlată cu exactitate.

Cu ajutorul limbajului JavaScript se pot descrie script-uri care pot modifica orice proprietate a layer-elor (le pot deplasa, ascunde, mări, micșora), practic pot modifica în mod dinamic conținutul paginii.

Există două modalități de definire a layer-elor:

1. cu ajutorul stilurilor (CSS);
2. cu ajutorul elementului HTML **LAYER**.

Definirea layer-elor cu ajutorul stilurilor CSS

Pentru a defini un layer cu ajutorul stilurilor, se definește un stil individual, care va fi asociat prin intermediul atributului **ID** elementului HTML, care constituie stratul respectiv. Dacă dorim ca un strat să fie alcătuit din mai multe elemente HTML, acestea trebuie să fie grupate cu ajutorul elementului **DIV** sau al elementului **SPAN**. Când definim un layer, trebuie să precizăm în primul rând poziția sa cu ajutorul proprietății **POSITION** ce poate lua două valori: **absolute** (ceea ce indică poziția absolută a colțului său din stânga-sus față de zona de vizualizare a documentului sau a layer-ului în care este eventual inclus), **relative** (ceea ce indică specificarea poziției în mod relativ în raport cu poziția pe care conținutul layer-ului ar fi avut-o în mod normal în document). Exemple:

```
<STYLE TYPE="text/css"> #layer1
    {POSITION: absolute; TOP: 200px; LEFT: 250px;}
</STYLE>
<STYLE TYPE="text/css"> #layer2
    {POSITION: relative; TOP: 20px; LEFT: 50px;}
</STYLE>
```

În cele două exemple, proprietățile **LEFT** și **TOP** specifică, în cazul poziției absolute, distanța pe orizontală și respectiv pe verticală colțului stânga-sus al layer-ului față de colțul stânga sus al documentului sau al layer-ului în care e inclus, respectiv distanța colțului layer-ului față de poziția sa normală în document, în cazul poziției absolute.

Alte proprietăți specifice layer-elor ce pot fi specificate cu ajutorul CSS sunt:

WIDTH – specifică lățimea layer-ului (în pixeli sau în procente relativ la dimensiunea zonei de vizualizare a documentului sau a layer-ului în care este inclus). Dacă layer-ul conține elemente care nu pot fi încadrate în lățimea precizată (de exemplu, o imagine), el este dimensionat corespunzător. Dacă nu este precizată lățimea layer-ului, atunci se consideră implicit până la marginea din dreapta a zonei de vizualizare sau a layer-ului în care e inclus.

HEIGHT – specifică înălțimea layer-ului (în pixeli sau în procente). Dacă înălțimea nu este suficientă

pentru a vizualiza conținutul layer-ului, acesta este redimensionat corespunzător. Specificarea înălțimii layer-ului este utilă atunci când layer-ul include alte layer-e, a căror dimensiune se specifică relativ la el. **CLIP: rect(stanga, sus, dreapta, jos)** – definește coordonatele colțurilor stânga-sus și dreapta-jos ale zonei dreptunghiulare care încadrează layer-ul, exprimate în număr de pixeli. Dacă lipsește acest atribut, colțul stânga-sus este definit de **LEFT** și **TOP**, iar colțul dreapta-jos se calculează în funcție de **WIDTH** și **HEIGHT**. Dacă și aceste atribute lipsesc, colțul din stânga-sus are implicit coordonatele (0,0), lățimea este definită de distanța până la marginea din dreapta a zonei de vizualizare a documentului sau a layer-ului în care este inclus, iar înălțimea este cea necesară pentru vizualizarea conținutului layer-ului.

Z-INDEX – specifică poziția layer-ului pe axa OZ. Mai exact, prin intermediul acestui atribut i se asociază fiecărui layer un număr natural care determină ordinea de suprapune a layer-elor: un layer cu număr mic este plasat sub un layer cu număr mai mare. Prin intermediul acestui atribut, practic se poate modifica ordinea implicită a layer-elor (ordinea definirii acestora).

VISIBILITY – specifică dacă layer-ul este sau nu vizibil. Atributul admite valorile: **show** (layer-ul este vizibil), **hide** (layer-ul nu este vizibil, adică este „ascuns”, dar el este prezent în document), **inherit** (layer-ul este vizibil sau, dacă este inclus într-un alt layer, are aceeași vizibilitate cu a layer-ului în care este inclus (o „moștenire”); aceasta este valoarea implicită).

BACKGROUND-COLOR – determină culoarea scrisului și culoarea de fundal pentru conținutul layer-ului. Implicit, un layer este transparent, în sensul că un eventual layer plasat desdesubt este vizibil prin zonele libere ale textului sau ale altor elemente HTML. Culoarea nu va fi aplicată la întreaga zonă dreptunghiulară a layer-ului, ci strict pentru text. Dacă vom descrie un chenar pentru layer (chiar invizibil), culoarea va fi aplicată uniform, în întreaga zonă a layer-ului.

BACKGROUND-IMAGE: URL (adresă_fișier_image) – specifică adresa fișierului ce conține o imagine ce va fi utilizată ca imagine fundal.

Definirea layer-elor cu ajutorul elementului LAYER

Elementul HTML **LAYER** permite descrierea unui layer în mod absolut.

<LAYER attribute> continutul layer-ului </LAYER>

Acest element admite atributele: **ID, TOP, LEFT, WIDTH, HEIGHT, CLIP, VISIBILITY, Z-INDEX**, cu aceeași semnificație ca în cazul stilurilor CSS. Atributul **BGCOLOR** corespunde proprietății

BACKGROUND-COLOR, iar atributul **BACKGROUND** proprietății **BACKGROUND-IMAGE**. Spre deosebire de **BACKGROUND-COLOR**, atributul **BGCOLOR** utilizează culoarea de fundal în întreaga zonă dreptunghiulară a layer-ului. De asemenea, atributul **BACKGROUND** determină multiplicarea imaginii de fundal în întreaga zonă a layer-ului, spre deosebire de **BACKGROUND-IMAGE**, care multiplică imaginea de fundal numai în porțiunea în care este vizualizat conținutul layer-ului.

În plus, elementul **LAYER** admite următoarele atribute:

SRC=adresă_fișier – specifică adresa unui fișier extern care descrie conținutul layer-ului. Acest atribut poate fi deosebit de util când informațiile afișate de layer se schimbă frecvent (în acest caz, se fac modificări doar în fișierul care descrie conținutul layer-ului) sau când dorim să prezentăm în layer un conținut dinamic (deoarece fișierul extern poate să conțină scripturi JavaScript).

ABOVE=nume_layer – specifică layer-ul plasat imediat deasupra layer-ului nou creat (acesta trebuie să existe deja).

BELOW=nume_layer - specifică layer-ul plasat imediat sub layer-ului nou creat (acesta trebuie să existe deja).

OnMouseOver, OnMouseOut, OnFocus, OnBlur, OnLoad – permit asocierea unui script care se execută la apariția evenimentului corespunzător (vezi utilizarea script-urilor JavaScript).

Elementul **ILAYER**, spre deosebire de elementul **LAYER**, permite descrierea unui layer poziționat relativ față de poziția pe care ar avea în mod natural în document.

<ILAYER attribute> continutul layer-ului </ILAYER>

Ca și în cazul elementului **LAYER**, pentru poziționare se utilizează atributele **LEFT** și **TOP**.

Elementul **NOLAYER** se utilizează în cazul în care un document care conține layer-e este vizualizat cu ajutorul unui browser care nu admite această facilitate, în acest caz conținutul layer-ului este vizualizat fără poziționare. Dar dacă documentul conține și scripturi JavaScript care utilizează layer-ele, se vor produce erori. Pentru a evita aceste erori, se poate utiliza elementul HTML **NOLAYER** (specificând un conținut alternativ între eticheta de început și cea de sfârșit, astfel:

<NOLAYER> continut_alternativ </NOLAYER>

De exemplu:

<NOLAYER>

Pentru vizualizarea acestei pagini este recomandat un browser care suporta layer-e.

</NOLAYER>

Utilizarea fonturilor

Sunt situații în care fonturile utilizate de autorul documentului HTML nu se găsesc pe calculatorul utilizatorului, în acest caz browser-ul înlocuindu-le cu un font implicit ceea ce compromite, uneori destul de grav, aspectul pagii. Browser-ele permit încorporarea în documentele HTML a fișierelor ce conțin fonturile, astfel încât să poată fi descărcate pe calculatorul utilizatorului asemenea unei imagini. Fonturile sunt descărcate în cache-ul browser-ului, astfel că ele sunt pe calculatorul utilizatorului doar atâta timp cât pagina Web respectivă este în cache. Prin urmare, utilizatorii care vizualizează pagina nu își pot copia fonturile respective pentru a le utiliza ulterior, astfel că drepturile de autor asupra fonturilor utilizate nu sunt periclitare.

Pentru a încorpora un font într-o pagină Web, acesta trebuie să fie deja instalat pe calculatorul celui care crează pagina. De asemenea, autorul paginii, trebuie să aibă și un utilitar² care să-i permită crearea unui **fișier de definire a fonturilor**. Cu ajutorul utilitarului, prin meniurile sale, se încarcă fonturile necesare paginii, apoi se salvează într-un fișier (cu extensia .pfr). Apoi, în documentul HTML se crează o legătură cu fișierul respectiv prin intermediul elementului HTML LINK sau cu ajutorul stilurilor CSS. De exemplu, dacă fișierul de definire a fonturilor este exemplu.pfr, cele două moduri de încărcare vor fi:

```
<LINK REL=FONTDEF SRC=" exemplu.pfr">
```

sau

```
<STYLE TYPE="text/css">
```

```
  @fontdef url(exemplu.pfr);
```

```
</STYLE>
```

După încorporarea fișierului de definire a fonturilor, fonturile respective pot fi utilizate oriunde în document (de exemplu, ca valoare pentru atributul **FACE** al elementului HTML **FONT** sau pentru proprietatea CSS **FONT-FAMILY**).

Programe (scripturi) CGI (Common Gateway Interface – Interfață comună pentru Porți de Acces)

Este o tehnică prin care autorii de pagini Web pot crea pagini cu adevărat interactive, care să permită schimburi de informații cu utilizatorul. CGI reprezintă de fapt programe stocate pe server care au rolul de a prelucra informațiile transmise de utilizatori prin intermediul formularelor, iar, în unele situații programul generează documente HTML de răspuns.

Programul CGI poate fi scris în orice limbaj de programare care permite citiri și scrieri. Cele mai utilizate limbaje pentru CGI sunt: Perl, TCL, JavaScript, VBScript, C/C++. Pentru utilizarea lor trebuie să contactați administratorul de sistem (WebMaster) pentru că, din motive de securitate a serverului, este posibil să nu vă permită instalarea și utilizarea lor. De obicei sunt preferate alte tehnici, cum ar fi PHP sau ASP (Active Server Page).

Dacă metoda de transmitere a informațiilor către server este **POST (METHOD=POST)**, atunci informațiile sunt transmise către dispozitivul standard de intrare, prin urmare pot fi citite în programul CGI cu ajutorul funcțiilor/procedurilor de citire specifice limbajului folosit.

Metoda **GET (METHOD=GET)** este metoda implicită, iar în acest caz informațiile transmise sunt atașate adresei URL, obținându-se un **URI**. Spre deosebire de **URL (Uniform Resource Locator)**, care conține informațiile necesare pentru localizarea unei resurse Internet, **URI (Uniform Resource Identifiers)** conțin și informațiile asociate tranzacției **HTTP** curente. De exemplu, presupunând că utilizați motorul de căutare <http://www.google.ro> și introduceți drept cuvinte cheie de căutare **Scripturi CGI**, poate fi generat următorul URI: https://www.google.ro/?qws_rd=ssl#q=Scripturi+CGI Prin urmare, acționând butonul Search ați transmis informații către programul **query**, localizat pe serverul www.google.ro în directorul **cgi-**

² Există diferite utilitare de definire a fonturilor, ca, de exemplu, *Typograph* (produs de firma HexMac și poate fi obținut de la adresa <http://hexmac.com>), sau *Font Composer Plugin for Communicator*.

bin. Informațiile transmise sunt precizate după caracterul ?, sub forma unor perechi: **câmp=valoare**.

Observăm că spațiile sunt înlocuite cu caracterul +.

Deși metoda **GET** este cea implicită, cea mai frecvent utilizată este metoda **POST**, datorită avantajelor pe care le are în raport cu **GET**:

- Volumul de informații care pot fi transmise prin metoda **POST** poate fi foarte mare, în timp ce prin metoda **GET** informațiile transmise sunt memorate într-o variabilă de mediu denumită **QUERY-STRING**;
- Pe serverul *Web* se găsește cu siguranță un fișier în care se înregistrează tranzacțiile **HTTP** când transmiteți prin metoda **GET** informații private (de exemplu, utilizați un formular de modificare a parolei, prin care transmiteți vechea și noua parolă), acestea vor fi înregistrate în fișierul respectiv, deci pot fi vizualizate de alți utilizatori;
- Informațiile transmise către server prin metoda **GET** sunt vizualizate în zona Location a browser-ului, efect neplăcut care nu apare în cazul metodei **POST**.

Pe parcursul tranzacțiilor **HTTP** sunt memorate informații despre client și server în niște variabile speciale, denumite variabile de mediu (*environment variables*). Pe categorii, acestea sunt:

- a) Variabile ce nu depind de tipul cererii emise de client, fiind setate pentru orice tip de cerere:

SERVER_SOFTWARE – reține numele și versiunea programului server, în formatul **nume/versiune**;

SERVER_NAME – reține adresa **IP** sau denumirea **DNS** a calculatorului gazdă pe care rulează programul server;

GATEWAY_INTERFACE – reține specificația **CGI** acceptată de server, în formatul **CGI/versiune** (de exemplu CGI/1.0);

- b) Variabile de mediu care depind de tipul cererii **HTTP**:

SERVER_PROTOCOL – reține numele și versiunea protocolului, în formatul **protocol/versiune**;

SERVER_PORT – reține numărul portului prin care a fost transmisă cererea (uzual 80);

REQUEST_METHOD – reține metoda prin care a fost transmisă cererea **HTTP**;

QUERY_STRING - reține șirul de caractere **ASCII** care urmează după ? în **URI**-ul programului **CGI**;

CONTENT_TYPE – reține tipul informațiilor transmise (pentru metodele **POST** și **GET**) sub forma **tip/subtip**;

CONTENT_LENGTH – reține lungimea datelor atașate (în număr de octeți);

PATH_INFO – reține informații suplimentare despre cale, furnizate de client (de obicei calea este specificată relativ);

PATH_TRANSLATED – reține varianta specificată în mod absolut pentru calea reținută în **PATH_INFO**;

SCRIPT_NAME – reține calea, specificată relativ, către programul **CGI** care va fi rulat de către server;

REMOTE_ADDR – reține adresa **IP** a calculatorului care emite cererea;

REMOTE_HOST – reține numele **DNS** al calculatorului care emite cererea (dacă serverul nu cunoaște acest nume, această variabilă rămâne nesetată, utilizându-se valoarea memorată în **REMOTE_ADDR**;

AUTH_TYPE – reține protocolul de autentificare a utilizatorului care solicită acces la programul **CGI** (dacă serverul este configurat pentru a permite autentificarea utilizatorilor);

REMOTE_USER – reține numele utilizatorului în cazul în care se realizează autentificarea.

- c) Variabile de mediu suplimentare, preluate din antetul cererii emise de client:

HTTP_ACCEPT – reține tipurile **MIME** pe care le acceptă browser-ul, în formatul **tip/subtip** și sunt separate prin caracterul virgulă(,);

HTTP_USER_AGENT – reține programul browser în formatul **program/versiune**.

Scripturi PHP