

CHESS PLAYERS

LUCRARE PENTRU ATESTAREA
COMPETENȚELOR PROFESIONALE

Elev: Bobos Anamaria

Profesor îndrumător :Butyka Edmond

Liceul :Teoretic Lucian Blaga

Cuprins

- Motivul alegerii temei

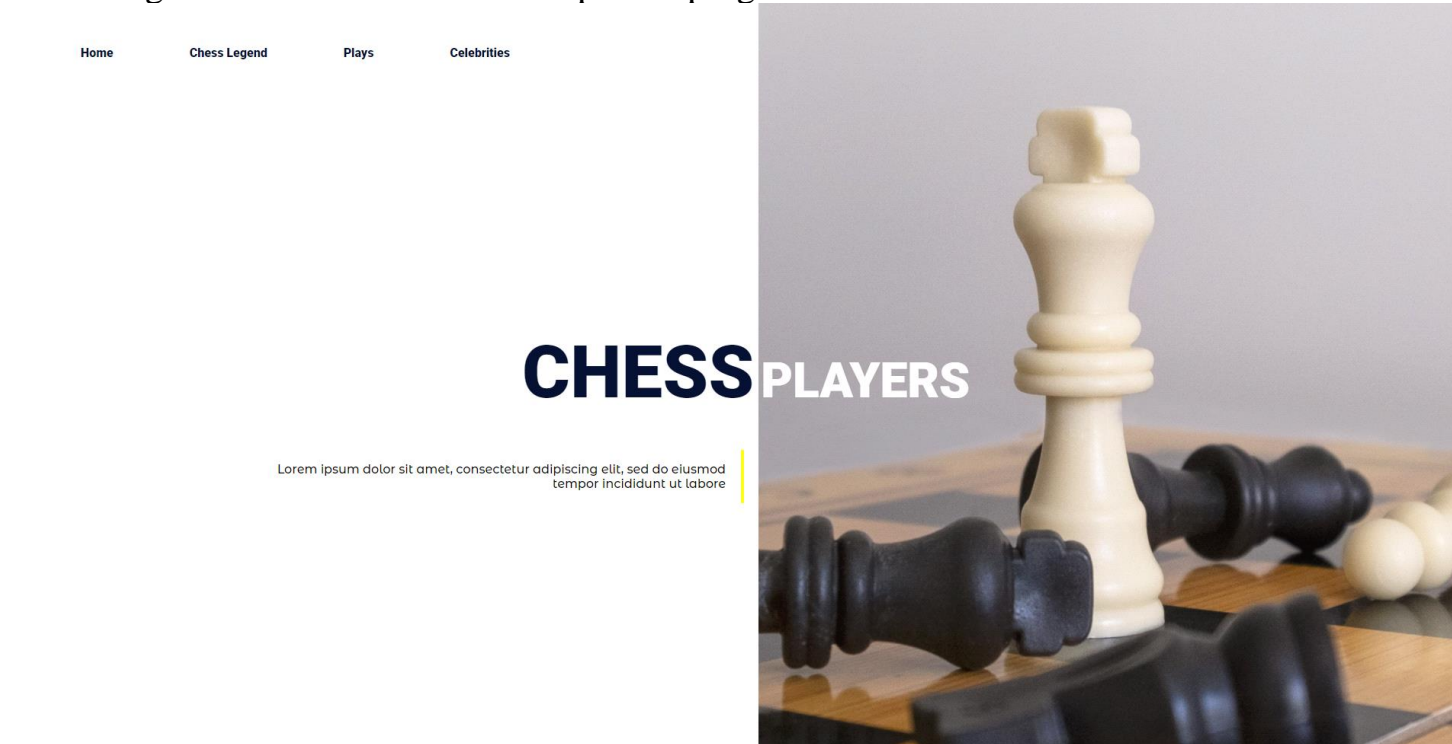
Șahul reprezintă sportul minții. Este una din pasiunile mele și consider că e important pentru cât mai

mulți oameni să învețe despre originile jocului de șah și să vadă frumusețea și dedicarea cu care oamenii joacă acest joc.

- Structura aplicației

Programul conține o formă principală 4 pagini: Home, Chess Legend, Plays și Celebrities.

Pagina Home ține locul unei coperte a programului.



Pagina Chess Legend conține un material video în care este povestită legenda șahului.

CHESS

LEGEND

[Home](#)[Chess Legend](#)[Plays](#)[Celebrities](#)

Pagina Plays conține o galerie de materiale video cu cele mai celebre partide de șah.

[Home](#)[Chess Legend](#)[Plays](#)[Celebrities](#)

Bobby Fischer vs Donald Byrne, 1956

The Game of the Century is a famous chess game that was won by the 13-year-old future world champion Bobby Fischer against Donald Byrne in the Rosenwald Memorial Tournament at the Marshall Chess Club in New York City on October 17, 1956.

In Chess Review, Hans Kmoch dubbed it 'The Game of the Century' and wrote: 'The following game, a stunning masterpiece of combination play performed by a boy of 13 against a formidable opponent, matches the finest on record in the history of chess prodigies.'

In this game, Fischer (playing Black) demonstrates noteworthy innovation and improvisation. Byrne (playing White), after a standard opening, makes a seemingly minor mistake on move 11, losing a tempo by moving the same piece twice.

Fischer pounces with brilliant sacrificial play, culminating in a queen sacrifice on move 17.

Byrne captures the queen, but Fischer gets far too much material for it – a rook, two bishops, and a pawn.

At the end, Fischer's pieces coordinate to force checkmate, while Byrne's queen sits useless on the other side of the board.



Anatoly Karpov vs Garry Kasparov



Bobby Fischer vs Donald Byrne



Garry Kasparov vs Veselin Topalov



Levon Aronian vs Visly Anand

Pagina Celebrities conține o listă cu cei mai celebri jucători de șah cum ar fi Garry Kasparov, Mikhail Botvinnik, José Raúl Capablanca, Bobby Fischer, Anatoly Karpov.



Garry Kasparov

Garry Kasparov, in full Garri Kimovich Kasparov, (born April 13, 1963, Baku, Azerbaijan, U.S.S.R.) is a Soviet-born chess master who became world chess champion in 1985. Kasparov was the youngest world chess champion (at 22 years of age).

Kasparov was born to a Jewish father and an armenian mother. He began playing chess at age 6, by age 13 was the Soviet youth champion and won his first international tournament at age 16 in 1979. Kasparov became an international grandmaster in 1980. From 1973 to 1978 studied under former world champion Mikhail Botvinnik.

Kasparov first challenged the reigning world champion Anatoly Karpov in a 1984–85 match, after he survived the Fédération Internationale des Échecs (FIDE; the international chess federation) series of elimination matches. Kasparov lost four out of the first nine games but then adopted a careful defensive stance, taking an extraordinarily long series of drawn games with the champion. With Kasparov finally having won three games from the exhausted Karpov, FIDE halted the series after 48 games, a decision protested by Kasparov. In the two player rematch in 1985, Kasparov narrowly defeated Karpov in a 24-game series and thereby became the youngest official champion in the history of the game. In 1996 Kasparov defeated a powerful IBM custom-built chess computer known as Deep Blue in a match that attracted worldwide attention. Kasparov and the team of Deep Blue programmers agreed to have a rematch in 1997. Deep Blue's intelligence was upgraded, and the machine prevailed. Kasparov resigned in the last game of the six-game match after 15 moves, granting the win to Deep Blue. In 2000 Kasparov lost a 16-game championship match to Vladimir Kramnik of Russia. Kasparov retired from competitive chess in 2005, though not from involvement in chess. In particular, he produced an acclaimed series of books, Kasparov on Great Predecessors (2003–06), that covered all the world chess champions from Wilhelm Steinitz through Karpov, as well as many other players.

- Detalii tehnice

Introducerea imaginilor se realizează utilizând tag-ul `` .

Pentru formatarea imaginilor și a paginii folosim limbajul CSS ce are ca scop stilizarea elementelor html.

Exemplu de așezarea în pagina a unui text și formatarea acestuia în limbajul

```
286 .textWrapper {
287     position: relative;
288     top: 20%;
289     left: 20px;
290     width: 70%;
291 }
292
293 .textP {
294     font-family: Montserrat Alternates;
295     font-style: normal;
296     font-weight: 300;
297     font-size: 15px;
298     line-height: 18px;
299     text-align: left;
300 }
```

CSS

Pentru a face deplasarea între pagini ușoară am **creat legături** între acestea prin intermediul unui navbar și a tagului `Text`

```
9     <nav>
10         <ul>
11             <li><a href="index.html">Home</a></li>
12             <li><a href="chess1.html">Chess Legend</a></li>
13             <li><a href="plays.html">Plays</a></li>
14             <li><a href="celeb.html">Celebrities</a></li>
15         </ul>
16     </nav>
```

Crearea unui navbar și interconectarea paginilor

JavaScript (JS) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbaj ce este folosit în proiect, pe pagina Plays pentru schimbarea videoclipurilor, încărcarea acestora în aceeași fereastră și afișarea informațiilor relevante în funcție de materialul video ales.


```
75     var video = document.getElementById('source');
76     var titlu = document.getElementById('titluMeci');
77     var text = document.getElementById('despreMeci');
78     switch (url.id) {
79         case "video/kvk.mp4":
80             titlu.innerHTML = title[0];
81             text.innerHTML = tex[0];
82             break;
83         case "video/bvd.mp4":
84             titlu.innerHTML = title[2];
85             text.innerHTML = tex[2];
86             break;
87         case "video/kvt.mp4":
88             titlu.innerHTML = title[1];
89             text.innerHTML = tex[1];
90             break;
91         case "video/lvv.mp4":
92             titlu.innerHTML = title[3];
93             text.innerHTML = tex[3];
94             break;
95         default:
96
97     }
98     video.src = url.id;
99     document.getElementById("video").load();
```

Codul sursă

- Limbajul HTML

HTML (Hypertext Markup Language) este un limbaj creat în scopul de a descrie, în mod text, formatul paginilor Web; fișierele create în acest limbaj vor fi interpretate de navigatoare, care vor afișa paginile în forma dorită (cu texte formate, liste, tabele, formule, imagini, hiperlegături, obiecte multimedia etc.). HTML a apărut ca o aplicație ISO standard (apartine standardului SGML - Standard Generalized Markup Language, specializat pentru hipertext și adaptat la Web).

Asa cum se poate deduce din numele limbajului, HTML descrie caracteristicile de format ale elementelor incluse prin procedee de marcare. Acestea pot fi asemănate intuitiv cu marcasele folosite în tipografie pentru a indica scrierea unui text cu un anumit tip de caractere. Fiecare element va fi introdus între două marcase ("tags", în limba engleză) - de început și sfârșit - (uzual) de forma `<marcaj> ... </marcaj>`. Caracterele speciale de delimitare a marcajelor "`<`", "`>`" permit deosebirea acestora de textul propriu-zis. De exemplu, pentru textele aldine (îngrosate), marcajul de început este `` iar de sfârșit - ``.

În informatică, limbajele de marcare sunt foarte convenabile fiindcă comenzile lor pot fi interpretate simplu. LaTeX-ul, de exemplu, este tot un limbaj de marcare; prin interpretarea fișierelor .tex descrise în acest limbaj se va genera formatul dorit al documentelor pe diverse tipuri de sisteme de calcul (în cazul, LaTeX-ului, se obține uzual format PostScript sau PDF). În schimb, procesoarele de documente uzuale nu au un limbaj de marcare standardizat, care să ofere compatibilitate între diverse tipuri de calculatoare și sisteme de operare. Astfel, se poate spune că avantajele aplicării limbajelor de marcare constau în portabilitate și flexibilitate: fișierele create cu ajutorul lor pot fi transferate pe orice tip de sistem, unde vor fi interpretate cu ajutorul unor programe specifice.

De fapt, procesoarele de texte uzuale folosesc adesea procedee de marcare pentru formătări (de exemplu, formătărilor de tip caracter din Word); în acest caz însă, caracterele de control introduse sunt ascunse iar rezultatul editării este direct vizibil ("What You See Is What You Get"). În schimb, în limbajele de marcare - inclusiv HTML - marcasele sunt introduse în text, astfel încât acestea sunt exclusiv succesiuni de caractere (litere, cifre, caractere speciale) - fișiere de tip text.

Referitor la legătura dintre procesoarele de documente uzuale și limbajul HTML, mai trebuie menționat faptul că ultimele versiuni ale editoarelor de

documente oferă facilități de salvare în format HTML - de exemplu, Word, începând cu versiunea Microsoft Office '97. Mai mult, toate produsele incluse în această gamă dedicată biroticii (MS Office) oferă compatibilitate cu formatul HTML.

Procese de standardizare și de includere a comenzilor de marcare în fișierele HTML permit navigatoarelor să citească și să formateze paginile Web, lucru foarte important în condițiile în care ele conțin nu numai texte alb-negru, ci și culori, imagini, hiperlegături, diverse obiecte. Practic, marcasele HTML asigură controlul asupra modului de afișare a obiectelor corespunzătoare în cadrul programelor de vizualizare a documentelor HTML - navigatoarele.

Limbajul HTML a evoluat în versiuni succesive, odată cu evoluția protocolului HTTP și a programelor de navigare. Astfel, HTML 1.0 era compatibil cu Mosaic, primul program de navigare, dar după apariția unor navigatoare noi, a fost necesară introducerea unui standard oficial Internet pentru construirea paginilor (HTML 2.0) și extinderea sa cu noi facilități: formule matematice, tabele, moduri avansate de descriere a organizării paginilor (începând cu HTML 3.0).

Standardizarea oficială a limbajului HTML a fost realizată de consorțiul WWW și dezvoltată de diverși producători de soft (unii dintre aceștia urmăresc chiar promovarea navigatoarelor proprii prin introducerea unor particularități în formatele oficiale).

Paginile HTML se pot crea cu orice editor de texte de către utilizatorii care cunosc limbajul HTML sau, mai simplu, se pot utiliza editoare speciale, în care obiectele se introduc interactiv iar codul HTML se generează automat. Având în vedere că și în acest caz este utilă cunoașterea marcajelor generate pentru corectarea eventualelor erori (mai ales în cazul link-urilor), vom prezenta în continuare entitățile care se pot introduce în paginile HTML și marcasele caracteristice acestora:

Elementele limbajului HTML

Toate obiectele HTML sunt introduse între marcase care le definesc; majoritatea acestora sunt de forma `<tip_obiect>` (la început) și `</tip_obiect>` (la sfârșit). Tipul standard al obiectului poate fi specificat cu majuscule sau minuscule; totuși, se recomandă utilizarea majusculilor fiindcă astfel marcasele ies în evidență.

Majoritatea navigatoarelor permit vizualizarea paginii curente în formatul sursă HTML (forma pe care o interpretează pentru afisarea paginii). La interpretare, programele de navigare ignoră spațiile și <Enter>-urile, aplicând formatarea specificată.

Există însă și marcaje cu parametri; acestea au forma <tip_obiect param1=valoare1 param2=valoare2 ...>. De exemplu, obiectele de tip imagine sunt introduse cu delimitatorul , care are diverși parametri. Pentru definirea hiperlegăturilor se folosește marcajul <A>, care are de asemenea parametri proprii. Codurile de marcarea HTML pot fi clasificate în următoarele categorii:

- marcaje de bază - cele care delimitează pagina / documentul HTML, titlul acestuia și corpul paginii;
- marcaje pentru structurarea documentului - care permit introducerea de subtitluri, paragrafe, linii de delimitare;
- marcaje pentru formatarea textului și crearea listelor;
- marcaje pentru crearea hiperlegăturilor (hyperlinks);
- marcaje pentru introducerea de obiecte - tabele, formule, imagini sau obiecte multimedia preluate din fișiere, formulare.

Vom descrie în paragrafele următoare, elementele caracteristice fiecăreia din aceste categorii.

Pentru structurarea și organizarea informațiilor din paginile web se pot utiliza frame-uri (marcajul <FRAME>), prin care la un moment dat se afișează mai multe ferestre conținând fiecare câte o pagină. Introducerea și gestiunea frame-urilor se realizează foarte convenabil folosind editoarele HTML.

Mai menționăm faptul că în ultimele versiuni ale limbajului HTML și ale browser-elor s-a introdus posibilitatea integrării, respectiv lansării în execuție prin navigator, a unor aplicații. Acestea sunt scrise în limbajul Java, un limbaj cu caracteristici distribuite și obiectuale, adaptat programării în Web; ele se numesc "applet"-uri și se introduc în sursele HTML cu marcajul <APP> sau <APPLET> . Ultimele versiuni de editoare HTML permit introducerea interactivă a applet-urilor Java.

Marcaje de bază

O pagină Web este delimitată de marcajele <HTML> și </HTML> care indică începutul și finalul documentului și conține:

- zonă de antet cuprinsă între marcatele: <HEAD> ... </HEAD> si
- un corp delimitat de marcatele: <BODY> ... </BODY>.

Zona de antet este utilizată de către programele de căutare pe site-urile web si permite specificarea titlului paginii care va fi afisat de navigator (nu apare propriu-zis în continutul paginii).

titlul, introdus în antet, este cuprins între marcatele <TITLE> ... </TITLE>.

Astfel, o formă extrem de simplă a unui document HTML ar putea fi cea din următorul exemplu:

```
<HTML>
<HEAD> <TITLE>Titlul documentului </TITLE>
</HEAD>
<BODY>Document HTML foarte simplu</BODY>
</HTML>
```

Marcaje pentru structurarea documentului

Programele de navigare asigură afisarea diferentiată a unor titluri si subtitluri pentru sectiunile paginii, după criteriile implementate în acest scop la conceperea sa (litere mai mari sau mai mici, diverse culori, litere aldine sau simple, unul sau mai multe rânduri libere după titlu etc.).

Titlurile de capitole sau subtitlurile sunt definite de marcatele <Hn>, unde n este o cifră între 1 si 6 care specifică nivelul titlului (1 este titlul principal iar 6 corespunde celui mai scăzut nivel). Astfel:

<H1> ... </H1> indică un subtitlu de nivelul 1

...

<H6> ... </H6> indică un subtitlu de nivelul 6.

Pentru separarea zonelor paginii se pot folosi treceri la:

- linie nouă - cu marcajul
, eventual cu desenarea unei linii orizontale - marcajul <HR>;
- paragraf nou - cu marcajul <P> (se inserează o linie nouă si eventual se face o indentare). Marcajul </P> desemnează sfârșitul de paragraf dar este mai rar folosit (se poate omite).

Marcaje pentru formatarea textului si crearea listelor

Formatările uzuale de texte permit scrierea cu caractere:

- aldine - în acest scop pentru acel text se aplică marcasele `... `;
- cursive (italice) - pentru acel text se aplică marcasele `<I> ... </I>`;
- subliniate - textul se introduce între marcasele `<U> ...</U>`.

Unele programe mai vechi de navigare nu permiteau reprezentarea acestor formate; în acest caz, se aplicau alte moduri de evidențiere a textelor respective (culori, video invers).

Alte tipuri de formatare de caractere care se pot defini se referă la:

dimensiunea fonturilor si culori - se va utiliza marcasele cu parametri:

` ... ` , care indică utilizarea unui font de dimensiune x si culoare y.

- x poate fi un număr între 1 si 12, indicând mărimea exactă, sau un număr cu semn, indicând mărimea relativă la dimensiunea curentă.
- y poate fi numele unei culori standard, specificată în limba engleză si între ghilimele, sau, eventual, descompunerea unei culori în componentele RGB, sub formă hexazecimală (câte două cifre hexa pentru fiecare componentă).

Suplimentar, se pot crea pagini de stiluri, în care să se definească stiluri logice, modificabile ulterior; în acest scop, se folosesc marcasele `<DN>` - definiție, `` - punere în evidență, `` - accentuare puternică.

Pentru tastaturile care suportă numai codurile ASCII, caracterele speciale se pot crea folosind formatul `&nume_caracter` - de exemplu, ``` pentru e. Având în vedere semnificația specială a caracterelor `<`, `>` si `&`, apariția lor în documente trebuie specificată sub forma caracterelor speciale: `<` , `>` , respectiv `&` ; .

Prezentăm în continuare un exemplu de document HTML, precum si modul în care acesta este afișat de Netscape Navigator.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

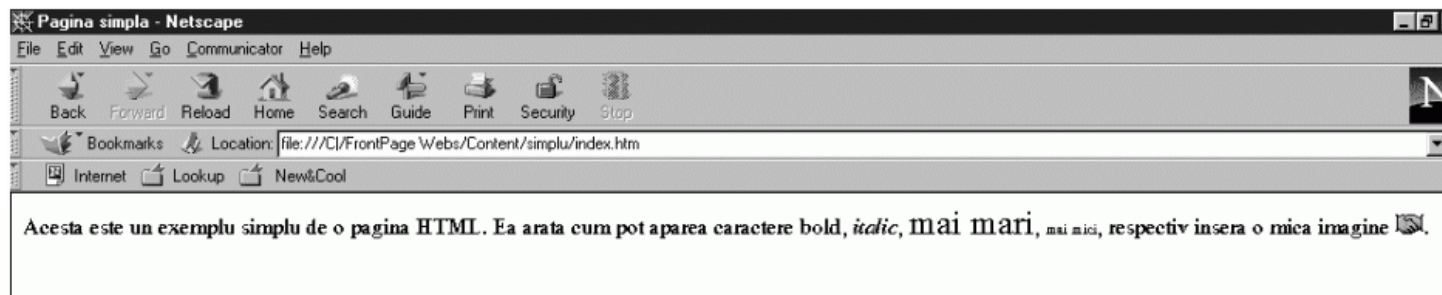
<head>
<title>Pagina simpla</title>
</head>
```

```

<body>
<p>Acesta este un exemplu simplu de o
pagina HTML. Ea arata cum
pot aparea caractere <strong>bold</strong>, <em>italic</em>,
<font size="5">mai mari</font>,
<font size="1">mai mici</font>,
respectiv insera
o mica imagine
.
</p>
</body>

</html>

```



În figura, se observă că textul propriu-zis al documentului poate fi delimitat față de codurile de marcare, iar formatarea textului din documentul HTML nu corespunde cu cea afișată de programul de vizualizare:

- textul este extins pentru a ocupa întregul ecran și este afișat cu un corp de literă diferit față de sursa HTML;
- terminatorii de linie nu sunt respectați;
- spațiile albe suplimentare din documentul sursă sunt ignorate.

Introducerea obiectelor de tip imagine (marcajul) este detaliată mai jos.

Limbajul HTML permite definirea mai multor tipuri de liste și imbricarea lor (includerea unor liste în altele), caz în care trebuie verificată cu atenție corespondența dintre marcasele de început și sfârșit pentru fiecare listă. Formatarea

listelor la afisarea paginii (introducerea bulinelor, numerotării etc.) se face de către programul de navigare.

Elementele listelor se introduc între marcasele ... ("list item"). Modul de aparitie al elementelor depinde de tipul de listă în care sunt incluse (cu buline, numerotate etc.), determinat de marcajul specific.

Astfel, listele pot fi:

- neordonate (cu buline) - delimitate de marcasele ... ("unordered list"); elementele lor se afisează cu buline;
- ordonate (numerotate) - delimitate de marcasele ... ("ordered list"); elementele lor se vor numerota;
- de tip meniu - delimitate de marcasele <MENU> ... </MENU>; elementele acestor liste apar într-o reprezentare compactă pe ecran (multicoloană);
- glosare (liste de definitii) delimitate de marcasele <DL> ... </DL> ("definition list"); elementele acestora au câte două componente, introduse respectiv cu marcasele <DT> (pentru nume) si <DD> (pentru semnificatia asociată);
- tabele scurte - create cu marcajul <DIR>.

Listele utilizate cel mai frecvent sunt cele ordonate si neordonate. Prezentăm în continuare un exemplu simplu de pagină HTML care contine aceste tipuri de liste.

```
<html>
```

```
<head>
```

```
<title>Pagina cu liste</title>
```

```
</head>
```

```
<body>
```

```
<p>Aceasta pagina exemplifica crearea listelor</p>
```

```
<p>Cele mai uzual folosite tipuri de liste sunt:
```

```
<ul>
```

```
<li>neordonate</li>
```

```
<li>ordonate</li>
```

```
</ul>
```

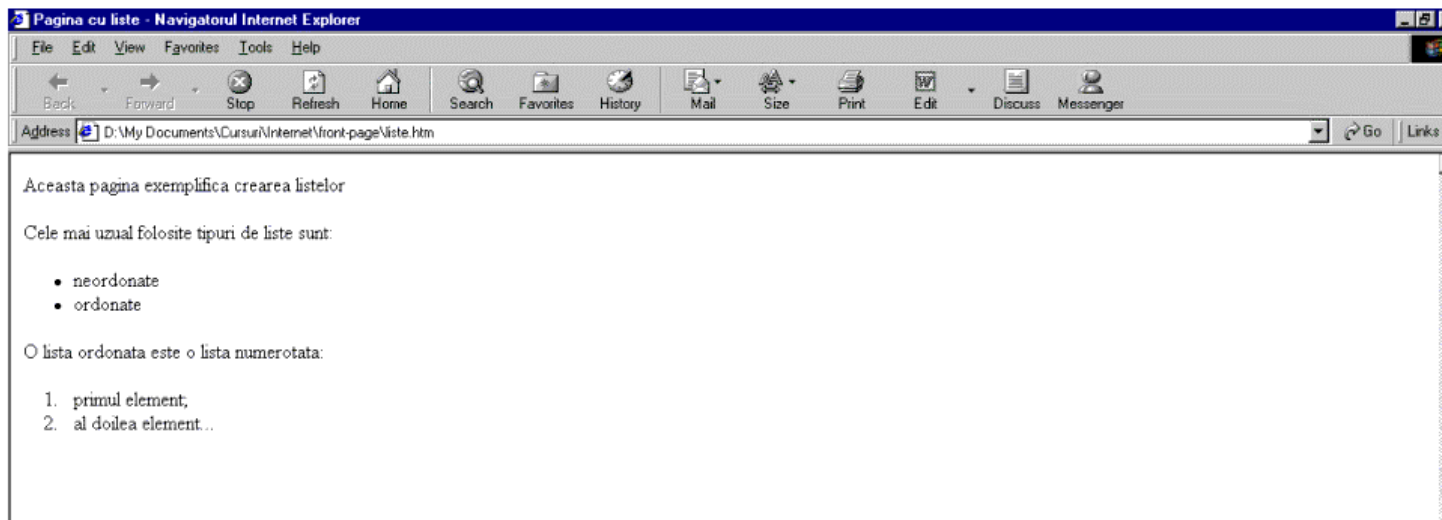

<p>O lista ordonata este o lista numerotata:

primul element;

al doilea element...

</body>

</html>



Exemple de liste într-o pagină HTML

Marcaje pentru crearea hiperlegăturilor

Hiperlegăturile ("hyperlinks") se introduc cu marcajele <A> ... ("anchor") aplicate asupra unui text sau a unei imagini. Ele se pot crea către:

- adrese URL - astfel se asigură accesul în cadrul procesului de navigare (prin protocoale specifice), nu numai la paginile web, ci și la diverse servicii Internet;
- fișiere / resurse locale adică aflate pe același calculator cu pagina creată - link-uri locale ; acestea ar putea fi accesate și printr-un URL cu protocolul "file" dar pentru mai multă simplitate, este suficientă specificarea numelui și căii fișierului local;
- o zonă din documentul HTML curent - link-uri interne.

În fiecare din aceste cazuri, un click pe textul sau imaginea respectivă va determina activarea legăturii și afisarea prin intermediul navigatorului a resursei asociate link-ului.

Resursa asociată unei legături este descrisă prin parametrii marcajului <A>:

- HREF - permite asocierea unei adresei URL, crearea unei legături locale prin specificarea numelui și căii fisierului corespunzător sau a unei legături interne documentului; este cel mai important parametru pentru construirea de hiperlegături;
 - NAME - numele hiperlegăturii sau al zonei dintr-o pagină către care se face referirea, în cazul unui link intern;
 - METHODS - metodele de acces (acest parametru se utilizează doar în prelucrări speciale).
- Astfel, un link local sau către o adresă URL se poate defini pe un text cu: text .

O legătură internă către o zonă a documentului curent se indică folosind marcajul .

Marcaje pentru introducerea de obiecte

Prima versiune de HTML (1.0) nu permitea descrierea tabelelor sau a informațiilor formate. Asemenea obiecte trebuiau create formatat și introduse ca obiect deja formatat, cu marcajele <PRE>, </PRE>. Aceste marcaje indicau navigatorului că textul inclus trebuia afișat caracter cu caracter, fără vreo intervenție de formatare. În versiunile ulterioare ale limbajului HTML, s-au introdus însă facilități elegante de formatare, de includere a tabelelor și a altor obiecte.

În HTML 3.0 s-au introdus obiectele de tip tabel, formate din linii orizontale și coloane verticale la a căror intersecție se formează celulele. Acestea pot conține intrări diverse: texte, figuri sau chiar alte tabele. Se pot realiza reuniri de celule (de exemplu, pentru titluri mai lungi). Atributele de formatare a tabelului (aliniere, stilul chenarului și marginilor etc.) se definesc în proiectarea paginii dar modul lor de afișare va depinde și de programul de navigare.

Tabelele se introduc între marcajele <TABLE> ... </TABLE>, căroră li se pot atașa (optional) parametrii BORDER și RULES. Un titlu pentru tabel se poate introduce cu marcajul <CAPTION>. Fiecare coloană se definește cu marcajul

<COL>, având ca parametru ALIGN - modul de aliniere a informațiilor din acea coloană (LEFT, CENTER, RIGHT).

Antetul tabelului se poate indica între marcasele <TH>...</TH> ("Table Header"), pentru trecerea la o linie nouă se utilizează marcasele <TR> ("Table Row") iar celulele individuale se marchează cu <TD> ("Table Data"), eventual cu parametru de aliniere. Se mai pot specifica alinieri orizontale sau verticale ale celulelor, grupări de celule etc. Aceste marcase permit navigatorului să afișeze diferentiat informațiile din tabel.

Prezentăm în continuare un exemplu simplu de tabel.

```
<HTML>
<HEAD>
<TITLE>Pagina cu tabel</TITLE>
</HEAD>
<BODY>
<H1>Aceasta pagina da un exemplu de folosire a unui tabel</H1>
<H3>Vanzari anuale:</H3>
<TABLE> <TH>1996</TH><TH>1997</TH><TH>1998</TH><TR>
<TD>125 milioane lei</TD><TD>160 milioane lei</TD><TD>231 milioane
lei</TD>
</TABLE>
</BODY>
</HTML>
```



Aceasta pagina da un exemplu de folosire a unui tabel

Vanzari anuale:

1996	1997	1998
125 milioane lei	160 milioane lei	231 milioane lei

Exemplu de tabel într-o pagină HTML

Mărimea liniilor și a coloanelor poate fi controlată suplimentar prin marcatele `<ROWSPAN=x>` și respectiv `<COLSPAN=y>`, fiecare indicând mărimea în "celule normale". În plus, se poate indica trasarea unei margini pentru tabel, prin marcajul `<TABLE BORDER>`.

Includerea imaginilor se face folosind marcajul ``, care are parametri specifici:

- **SRC** indică (după semnul '=') URL-ul imaginii, respectiv calea fișierului dacă acesta este local; uzual, se acceptă fișiere în format GIF sau JPEG;
- **ALIGN** controlează alinierea imaginii față de limita inferioară a textului (TOP, MIDDLE sau BOTTOM); este un parametru optional;
- **ALT** furnizează textul afișat în locul imaginii dacă utilizatorul dezactivează opțiunea de afișare a imaginilor (parametru optional);
- **ISMAP** este un indicator optional pentru imaginile care sunt hărți selectabile.

Astfel, introducerea simplă a unei imagini se poate realiza cu ``. Un exemplu de imagine inserată într-o pagină HTML este prezentat în figura de mai sus.

Remarcăm faptul că, utilizând parametri specifici, marcajul `` se poate folosi și pentru inserarea unor fișiere multimedia, cum ar fi secvențele video. Având în vedere că aceste operații se realizează mai ușor folosind un editor HTML.

Formulare. HTML 1.0 permitea uzual doar transferarea informațiilor de la furnizori către utilizatori, transferul în sens invers fiind foarte dificil. Odată cu dezvoltarea Web-ului și mai ales cu utilizarea sa în scopuri comerciale și în diverse alte domenii, s-a făcut simțită necesitatea comunicării în dublu sens pentru preluări de comenzi, completarea unor fișe de înregistrare, distribuirea de produse soft, administrarea de chestionar, transmiterea unor informații personale etc. Acestea au fost motivele introducerii formularelor în HTML 2.0.

Formularele conțin obiecte de control care permit utilizatorilor introducerea de informații prin completarea unor câmpuri specifice (casete de text), prin selectarea sau activarea unor opțiuni (comutatoare, grupuri sau liste de opțiuni). Aceste informații vor fi transmise, după activarea butonului SUBMIT (echivalentă cu tastarea lui `<Enter>`) proprietarului paginii, introduse în baze de date dedicate și prelucrate cu aplicații specifice.

Formularele se introduc prin marcăjelele <FORM> ... </FORM> , cărora li se atasează parametri specifici care definesc metodele de transmitere si tratare a datelor. Un formular poate contine obiecte de control de diverse tipuri, definite cu marcajul <INPUT> si o varietate de parametri care stabilesc, pentru obiectul definit, tipul, dimensiunea si modul lui de afisare. Astfel, se pot crea câmpuri de text (care vor fi completate cu texte de către utilizator), liste de alternative, comutatoare, grupuri de optiuni, butoane, hărți active etc.

Datele preluate prin intermediul formularelor se tratează conform standardului CGI (Common Gateway Interface) prin intermediul unor programe sau script-uri CGI. În plus, un asemenea script poate interactiona cu baza de date creată pentru realizarea unor actiuni specifice (de exemplu, hărțile active folosesc script-uri CGI pentru a executa diferite actiuni, în functie de zona selectată de utilizator).

Script-urile CGI pot executa si alte operatii decât prelucrarea formularelor, producând iesiri convenabile. Dacă o hiperlegătură indică spre un script CGI, la selectarea legăturii se va executa script-ul (cu anumite variabile de context care retin diverse informatii de stare). Script-ul va produce un fisier, de exemplu o pagină web, care va fi interpretat(ă) de navigator. Acest mecanism permite script-urilor să genereze, aproape instantaneu, pagini web care să satisfacă diverse cerinte ale utilizatorilor, furnizând anumite răspunsuri asteptate în urma unor actiuni.

Crearea interactivă a formularelor, folosind produse soft specializate în realizarea de pagini Web este relativ accesibilă pentru utilizatorii familiarizati cu caracteristicile obiectelor de control specifice aplicatiilor soft dar tratarea ulterioară a informatiilor transmise necesită cunostinte de specialitate.

Numele, simbolurile si procedura de introducere interactivă dintr-un editor HTML (de exemplu, Microsoft Front Page) a obiectelor de control sunt extrem de apropiate de cele asociate obiectelor similare din formularele sau rapoartele Microsoft Access. Pentru utilizatorii mai putin avizati însă, crearea, chiar interactivă, a formularelor este mai dificilă.

- Prezentarea programului în care a fost realizat proiectul

Atom este un editor de text și cod de sursă gratuit și deschis pentru macOS, Linux și Microsoft Windows cu suport pentru plugin-uri scrise în Node.js și Git Control încorporat, dezvoltat de GitHub .

Atom este o aplicație desktop construită folosind tehnologii web. Majoritatea pachetelor extinse au licențe software gratuite și sunt construite și întreținute în comunitate.

Atom se bazează pe Electron (anterior cunoscut sub numele de Atom Shell), un cadru care permite aplicațiile desktop de tip platformă multiplă folosind Chromium și Node.js. Este scris în CoffeeScript și Less.

Atom a fost eliberat din versiunea beta, ca versiunea 1.0, la 25 iunie 2015. Dezvoltatorii săi îl numesc „editor de text hackabil pentru secolul XXI”. Este complet personalizabil în HTML, CSS și JavaScript.

- Resurse de hard și soft necesare

Resurse minime :

1) Hardware :

- Procesor (CPU) : 500 MHz +
- Memorie (RAM): 64 MB
- Spațiu liber pe hard-disk : 30 MB
- Monitor :1024*768
- Un browser oarecare: Internet explorer, Mozilla, Opera

2) Software:

- Sistem de operare : Windows 98/XP

Resurse recomandate:

1) Hardware :

- Procesor (CPU) : 1000 MHz +
- Memorie (RAM): 128 MB
- Spațiu liber pe hard-disk : 30 MB
- Monitor : VGA 32-bit color (1024x768)
- Un browser oarecare: Internet explorer ,Mozilla
-

2) Software

- Sistem de operare : Windows XP