

Programare Web

Curs 2, 3

**Limbaajul HTML: elemente
de baza, formulare,
tabele, cadre**

Do you need to learn HTML?

- ◆ “...should a web designer need to be an HTML coder? Or can he just use a WYSIWYG tool?” (vezi [1])
- ◆ Raspunsul de la pagina de mai sus (si al meu) este Da.
- ◆ In cele ce urmeaza sunt cateva dintre argumentele de acolo.

HTML isn't rocket science-

- ◆ Go ahead and scoff. A lot of artists scoff when I tell them this. You might be thinking that it's easy for me to say this when I have almost over 11 years of HTML experience under my belt, but it's true. HTML is fairly easy to learn and I would estimate that 90% of what the average artist will do on their web page can be accomplished by only learning less than 20 HTML tags. Before you start your web page, get a good basic book on HTML like Head First HTML with CSS & XHTML (One of Kathy Sierra's great titles – [2]) or Sams Teach Yourself HTML & XHTML in 24 Hours and keep it around as a reference. Either one of these books will have you writing your own web pages within hours.

Learning HTML will give you greater control over the look of your site

- ◆ I generally use Dreamweaver make a web page quickly and to get the main structure of the page in place. I then go into the actual HTML code to tweak the design more precisely. Occasionally, you'll run into a situation in which your WYSIWYG tool is formatting something kind of strange and unless you can crack open the code, look at it and know what it's doing, you won't be able to make these small corrections very easily.

You'll be able to make changes no matter what tool you're using

- ◆ Say you're at a friend's house looking at your site and notice a big glaring error. If you know HTML, you'll be able to go directly to the control panel on your web host, open the page and make the correction right then and there. If you know HTML, you won't be tied a specific software and you won't be out of luck if something happens to your software or your computer. As long as you have access to a computer hooked to the internet, you'll always be able to make changes to your web site.

You aren't tied to a designer if you're having your site built for you

- ◆ Even if you're getting someone else to design your site, I still suggest that you learn basic HTML so you can maintain your site and add pages yourself. Even after your web site has been built, you'll still have a certain amount of changes and additions you'll need to make in the future. If you have to depend on your web designer to make these changes you're at the mercy of their schedule and availability and it will cost you extra to have them make these changes. If you know basic HTML and get them to design a simple template for you to follow when creating new pages, you'll be able to take full control of your web site.

Limbajul HTML

- ◆ Așa cum am văzut în primul capitol, serverul web livrează la cerere fișiere HTML.
- ◆ Acestea sunt fișiere text conținând și directive care sunt folosite de browser pentru afișarea pe ecran a textelor, imaginilor, tabelelor, formularelor, ferestrelor și a celorlalte elemente pe care cel care a compus fișierul le-a utilizat.
- ◆ Aceste directive le putem vedea doar dacă vizualizăm sursa documentului și de regulă nu apar pe ecran.

Limbaajul HTML

- ◆ Scopul acestui capitol nu este acela de a face o descriere a tuturor acestor directive. Există numeroase documente și cărți publicate inclusiv în limba română care conțin informații complete în domeniu.
- ◆ Prezentam în continuare doar o trecere în revista a limbajului cuprinzând directivele uzuale.

Generalitati

- ◆ În HTML pentru a diferenția directivele de restul elementelor acestea se pun între paranteze ascuțite:

`<directiva>`

- ◆ O mare parte a lor au efect asupra unei întregi porțiuni din fișier, încadrata de început – sfarsit :

`<directiva>`

.....

`</directiva>`

Format general

```
<html>
```

```
<head>
```

Directive conținând informații generale
despre pagina

```
</head>
```

```
<body>
```

Conținutul paginii incluzând texte și
directive

```
</body>
```

```
</html>
```

Hello world

```
<html>  
<head>  
<title>Hello world title</title>  
</head>  
<body>  
HELLO WORLD!  
</body>  
</html>
```

Sau, mai simplu:

```
HELLO WORLD!
```

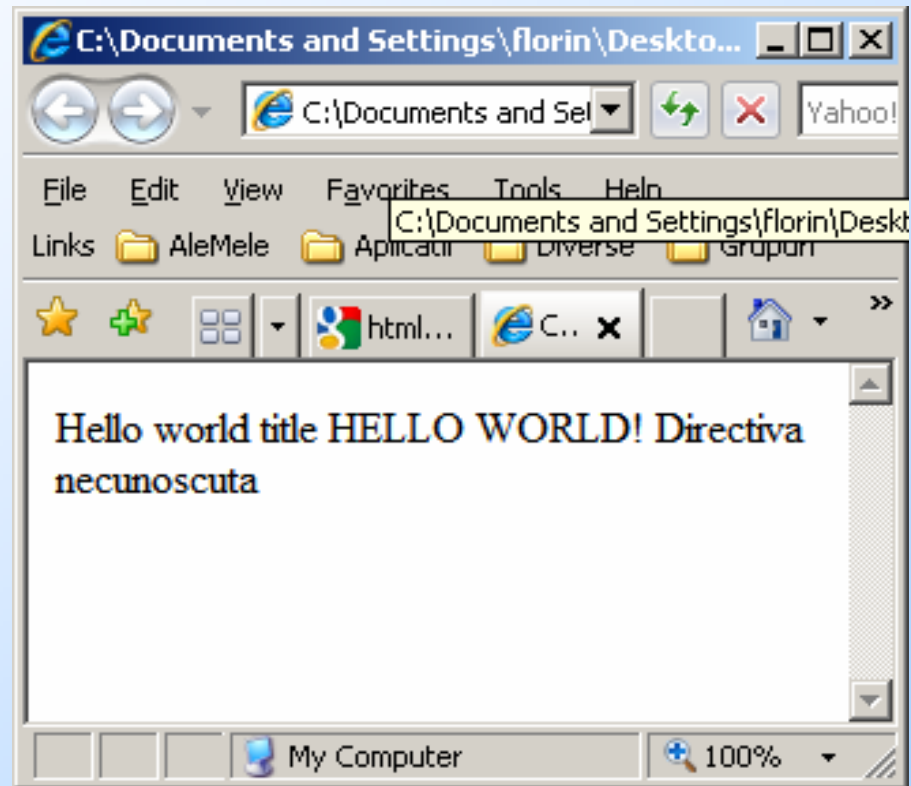
Erori

- ◆ Nu putem vorbi de semnalarea erorilor in HTML
- ◆ Daca un Browser un recunoaste o directiva pur si simplu o ignora.
- ◆ Exemplu:

```
<html>  
<hed>  
<turtle>Hello world title</turtle>  
</hed>  
<bady>  
<h 1>HELLO WORLD!</h 1>  
<mimi>Directiva <upb>necunoscuta</upb></mimi>  
</bady>  
</html>
```

Erori

- ◆ Rezultatul va fi:



Stiluri

In zona de continut a paginii (intre `<body>` si `</body>` putem folosi elemente ca:

- ◆ **Stiluri (headings):** Acestea sunt asemănătoare ca efect selectării în Microsoft Word a stilurilor Heading 1, Heading 2, etc.

Exemplu:

`<h1>` Titlu principal `</h1>`,

`<h2>` Titlu secundar `</h2>` ...

Liste neordonate

Lista fără numere de ordine (unordered list)

` `:

``

`Primul element în listă`

`Al doilea element în listă`

`.....`

`Ultimul element din listă`

``

Liste ordonate

Lista cu numere de ordine (ordered list) ` ...`
` ... `:

```
<ol>
```

```
<li>Primul element în listă
```

```
<li>Al doilea element în listă
```

```
.....
```

```
<li>Ultimul element din listă
```

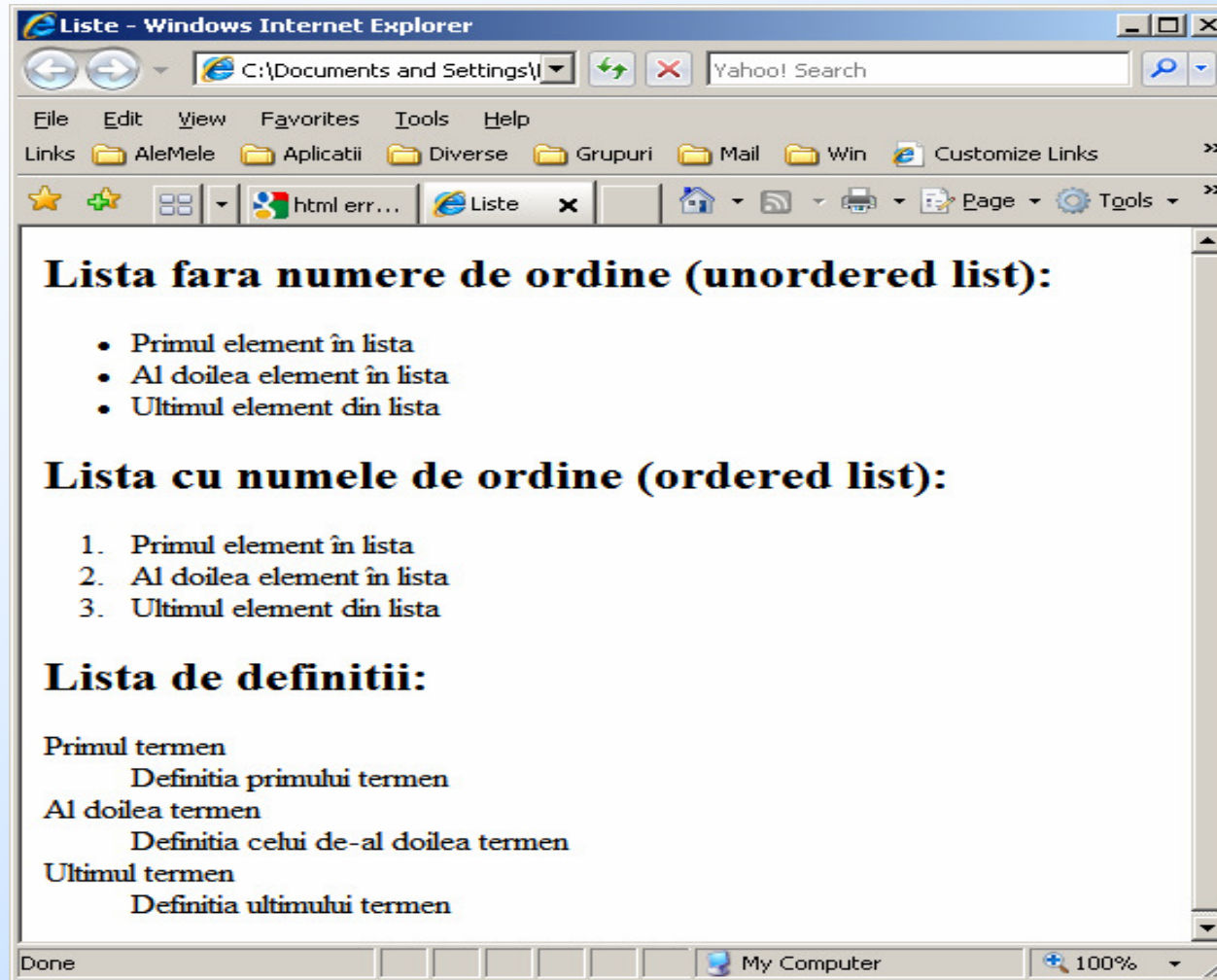
```
</ol>
```


Liste de definitii

Lista de definiții:

```
<dl>
  <dt>Primul termen</dt>
  <dd>Definiția primului termen</dd>
  <dt>Al doilea termen</dt>
  <dd>Definiția celui de-al doilea
    termen</dd>
  . . . . .
  <dt>Ultimul termen</dt>
  <dd>Definiția ultimului termen</dd>
</dl>
```

Exemplu

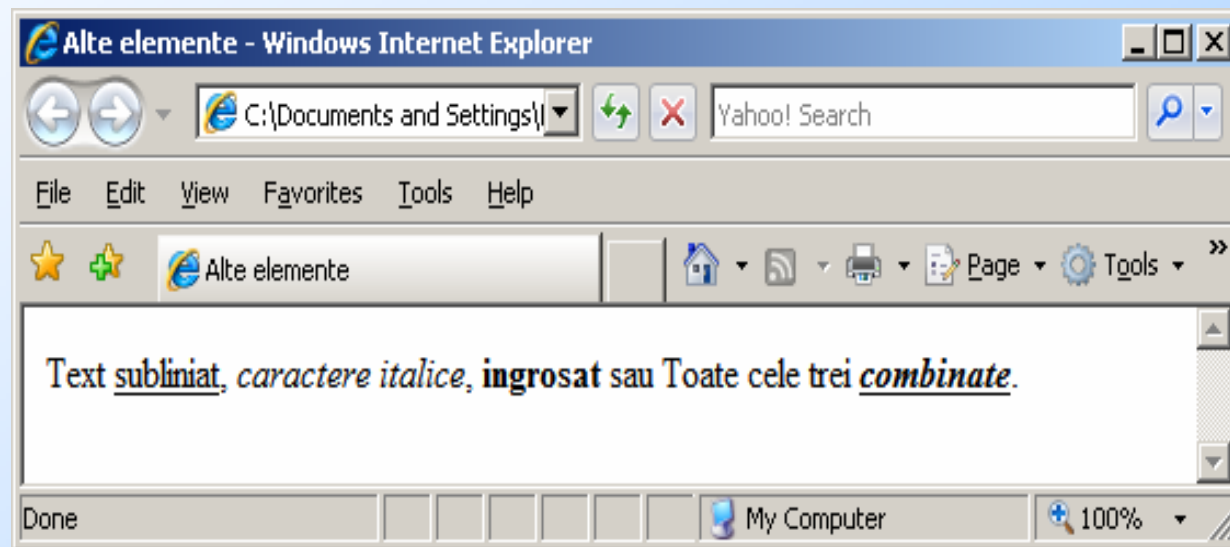


Alte directive de formatare

- ◆ Trecere pe linia următoare (break): `
`
- ◆ Linie orizontală (horizontal ruler): `<hr>`
- ◆ Paragraf: `<p> ... </p>`. Observație: `</p>` este facultativ.
- ◆ Subliniere: `<u> ... </u>`
- ◆ Italice: `<i> ... </i>`
- ◆ Îngroșare (bold): ` ... `

Exemplu

Text `<u>subliniat</u>`, `<i>caractere
italice</i>`, `ingrosat` sau
Toate cele trei
`<u><i>combinat</i></u>`.

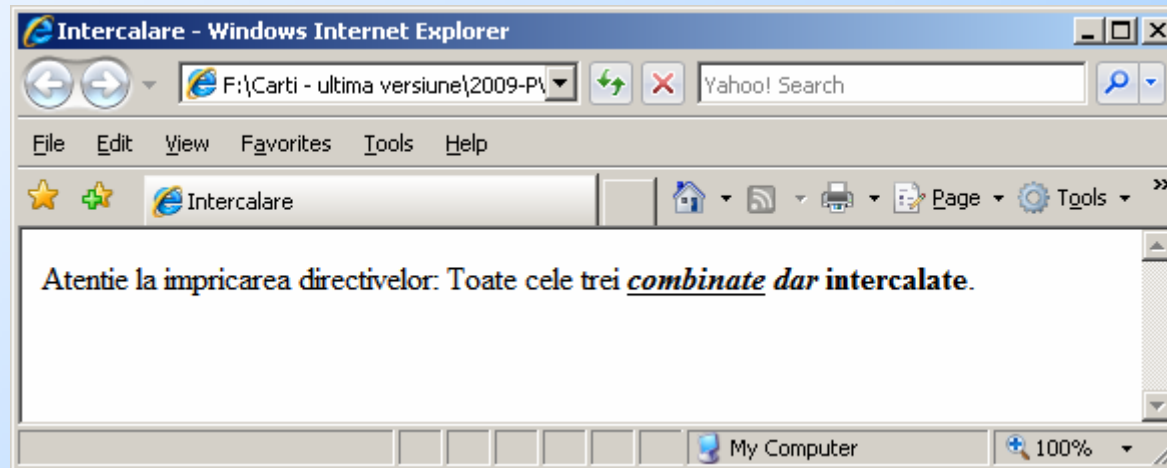


Exemplu

Fiecare pereche de directive de formatare isi face efectul independent de celelalte:

Toate cele trei

`<u><i>combinate</u> dar </i>
intercalate.`



Ancora (link)

- ◆ Acest tip de directivă este esențial și este cel care implementează de fapt conceptul de hipertext.
- ◆ În cadrul unei pagini HTML unul sau mai multe cuvinte consecutive pot să aibă asociată o altă pagină HTML.
- ◆ În momentul în care utilizatorul selectează (click) zona respectivă, automat browserul cere documentul asociat și îl afișează pe ecran.
- ◆ Forma cea mai simplă a unei astfel de directive este:

```
<a href="URL-adresa paginii asociate">
```

```
Cuvinte care se pot selecta
```

```
</a>
```

URL

Adresa specificată prin *href* poate fi:

- ◆ Relativă la pagina curentă (exemplele 1 și 2 de mai jos)

1. Exemplul 1: `Pagina 1
`
2. Exemplul 2: `Pagina 2
`

URL

Sau poate fi:

- ◆ Absolută, în care caz se specifică o cale completă pe același server de web (exemplul 3) sau o adresă completă (exemplul 4)

3. Exemplul 3: `Pagina 3
`

4. Exemplul 4: `Pagina 4
`

URL

- ◆ În primul caz noua pagină se găsește pe același server și în același director cu pagina curentă.
 - ◆ În al doilea caz, pentru găsirea paginii *pag2.html* serverul web pornește din directorul paginii curente și caută în subdirectorul *documente*.
1. Exemplul 1: `Pagina 1
`
 2. Exemplul 2: `Pagina 2
`

URL

- ◆ În al treilea caz, pentru găsirea paginii *pag3.html* serverul web pornește din rădăcina arborelui de documente (DOCUMENT_ROOT) și caută în subdirectorul *documente*.
- ◆ În ultimul caz noua pagină se poate găsi pe alt server de web decât cea curentă. Browserul va contacta acest server care îi va livra pagina aflată în subdirectorul *doc*.
- 3. Exemplul 3: `Pagina 3
`
- 4. Exemplul 4: `Pagina 4
`

Bookmark

Trimiteri în interiorul unui document:

- ◆ În acest caz marcarea locului referit se face cu:

```
<a name="marcaj-tinta">. . . </a>
```

- ◆ Referința are forma

```
<a href="adresa-document#marcaj-tinta"> . . . </a>
```

unde adresa documentului poate fi relativă sau absolută.

- ◆ Iată un exemplu cu două documente HTML în care primul reprezintă cuprinsul (tabla de materii) pentru al doilea.
- ◆ De remarcat că prin selectarea unei ancore care trimite spre o porțiune a unui document acesta este încărcat în întregime de browser dar poziționarea se face nu la începutul său ci în punctul țintă marcat.

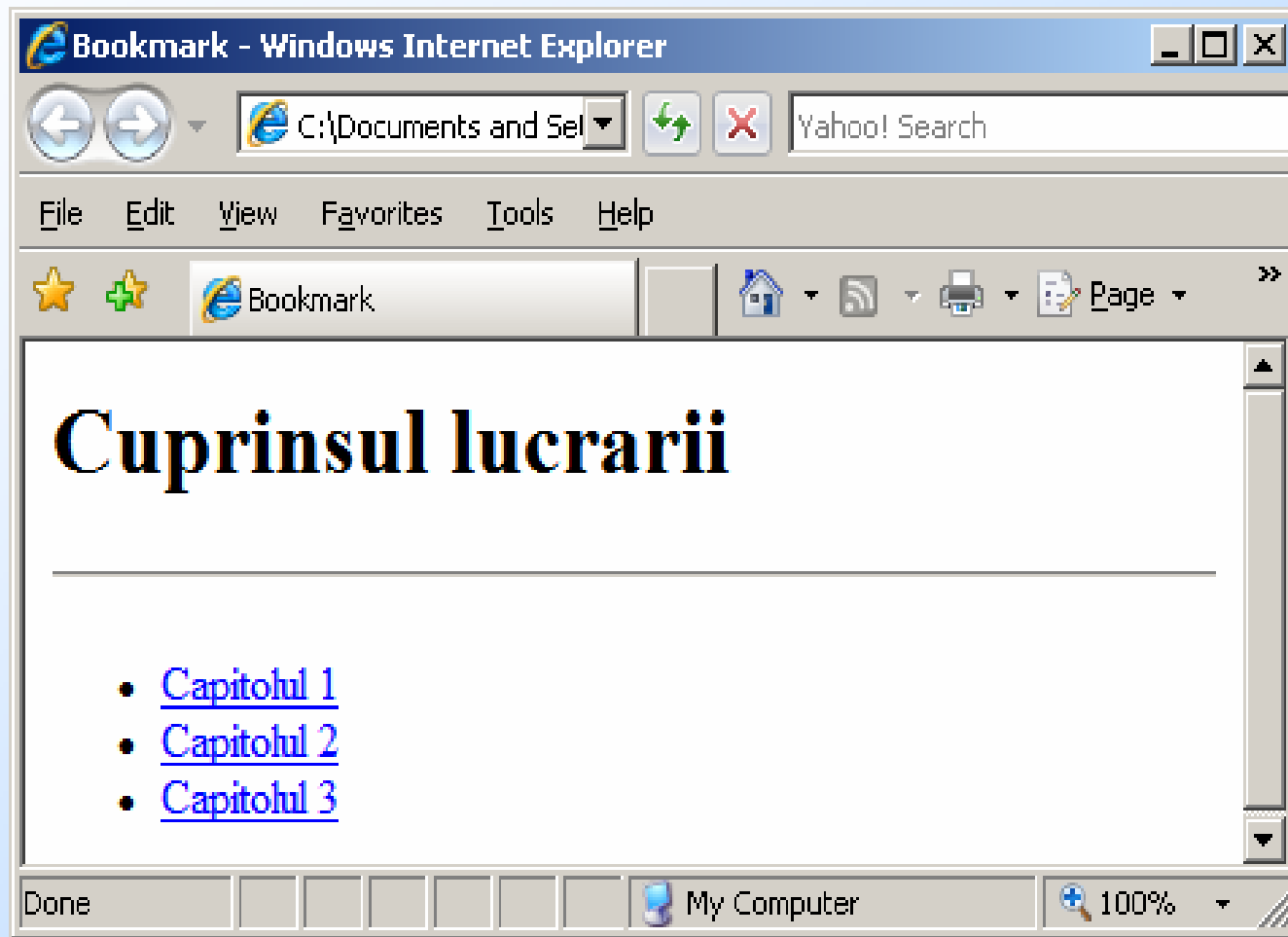
Exemplu - cuprins

```
<html>
<head>
<title>Cuprins</title>
</head>
<body>
<h1>Cuprinsul lucrarii</h1>
<hr>
<ul>
<li><a href="Continut.html#cap1">Capitolul 1</a>
<li><a href="Continut.html#cap2">Capitolul 2</a>
<li><a href="Continut.html#cap3">Capitolul 3</a>
</ul>
</body>
</html>
```

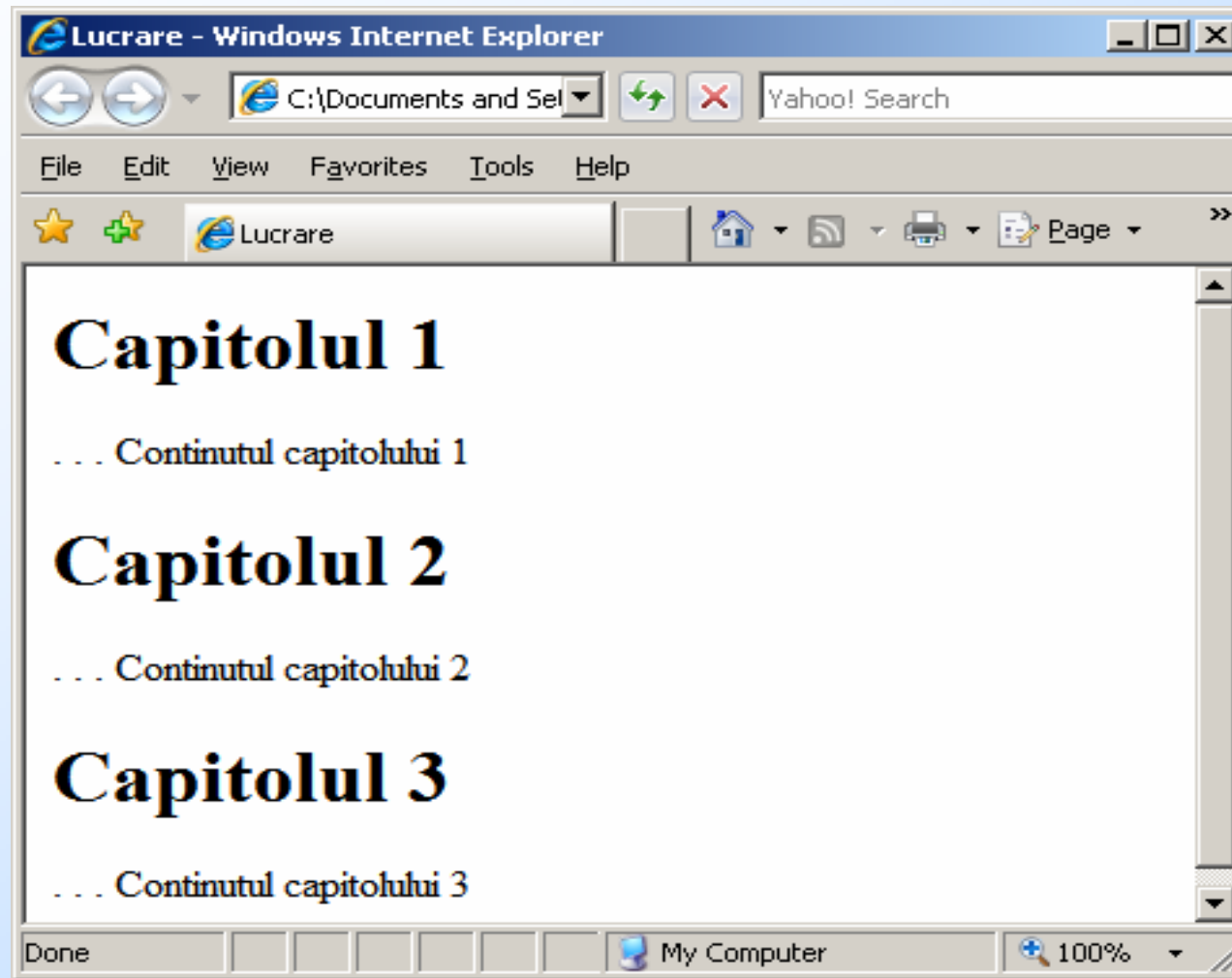
Exemplu - documentul

```
<html>
<head>
<title>Lucrare</title>
</head>
<body>
<h1><a name="cap1">Capitolul 1</a></h1>
. . . Continutul capitolului 1
<h1><a name="cap2">Capitolul 2</a></h1>
. . . Continutul capitolului 2
<h1><a name="cap3">Capitolul 3</a></h1>
. . . Continutul capitolului 3
</body>
</html>
```

Rezultat - cuprins



Rezultat - document



Imagini

- ◆ Documentele HTML pot conține directive de includere imagini (includerea imaginii in document se face la afisarea acestuia).

- ◆ Sintaxa (simplificata) este:

```

```

- ◆ *src* specifică adresa fișierului conținând imaginea (relativă sau absolută), *width* și *height* elemente de scalare a imaginii și *alt* un text alternativ pentru browserele care nu afiseaza imagini (textul poate apare si la pozitionarea pe imagine).

Imagini

- ◆ Fiecare tip de browser lucrează cu un anumit set de formate de imagine.
- ◆ Formatele JPEG și GIF sunt cele mai folosite dar nu sunt singurele.
- ◆ Subdirectivile *width*, *height* și *alt* sunt opționale.

Exemplu

```
  
Fotografia mea
```

Se observa deformarea
Imaginiii din cauza
lui width si height

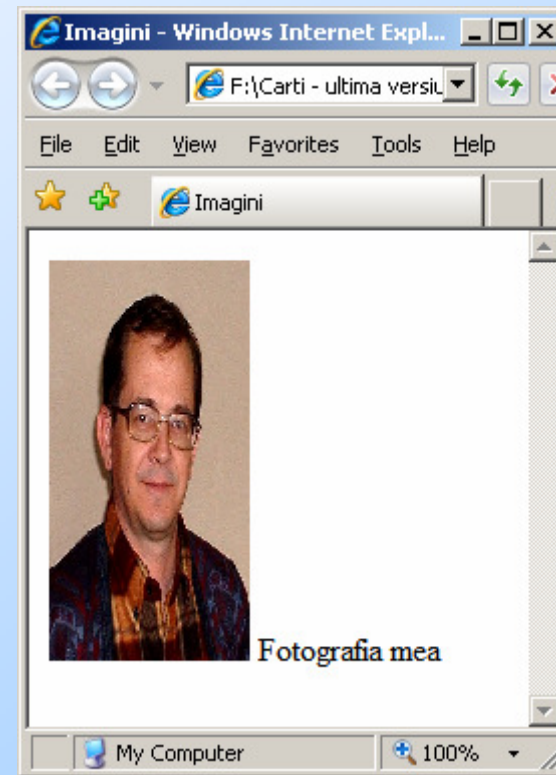


Image Link

- ◆ O imagine poate fi folosita la link:

```
<BODY>
```

```
<a href="test2.html"></a>
```

Fotografia mea

```
</body>
```

Border=0 elimina chenarul
plasat de browser automat
in jurul imaginii-link
(vezi dreapta).

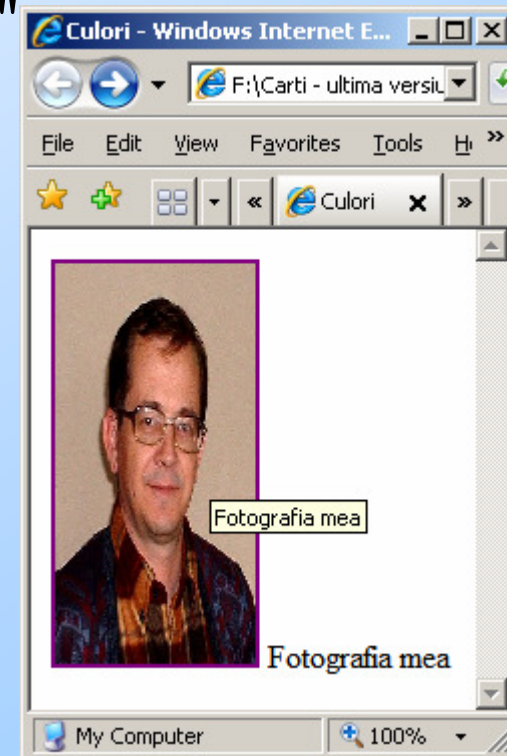


Image Link

- ◆ Link-ul poate cuprinde atat imagini cat si text:

```
<BODY>
```

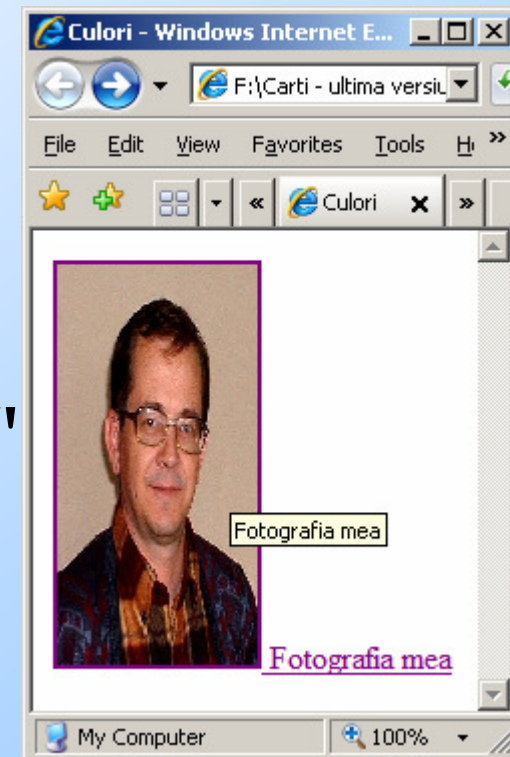
```
<a href="test2.html">
```

```

```

```
Fotografia mea</a>
```

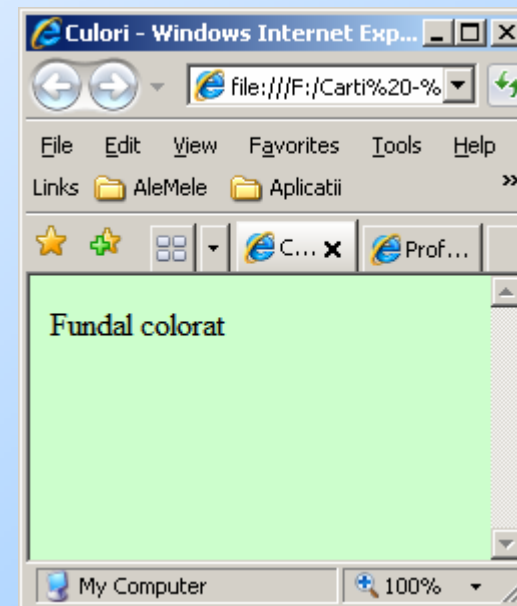
```
</body>
```



Background

- ◆ Se poate seta o culoare sau o imagine de fundal:

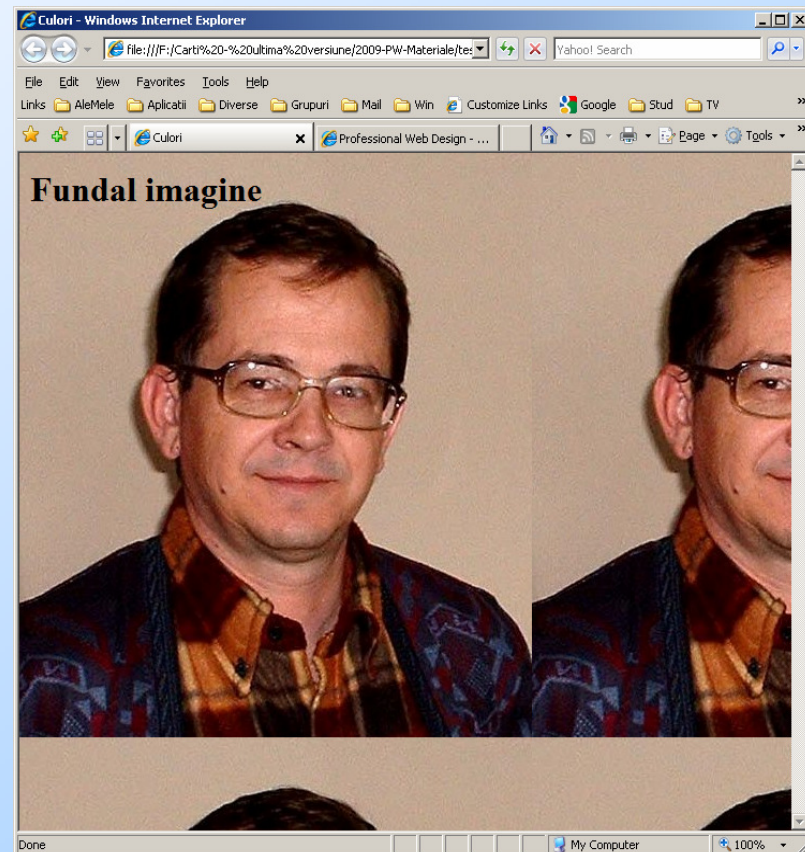
```
<html>
  <head>
    <title>Culori</title>
  </head>
  <body bgcolor="#CCFFCC">
    Fundal colorat
  </body>
</html>
```



Imagine de fundal

```
<html>
<head>
<title>Culori</title>
</head>
<body background=
  "foto2.jpg">
<h1>Fundal imagine</h1>
</body>

</html>
```



Linii orizontale

- ◆ Cu `<hr>` putem genera linii orizontale:

```
<BODY>
```

Prima parte

```
<HR>
```

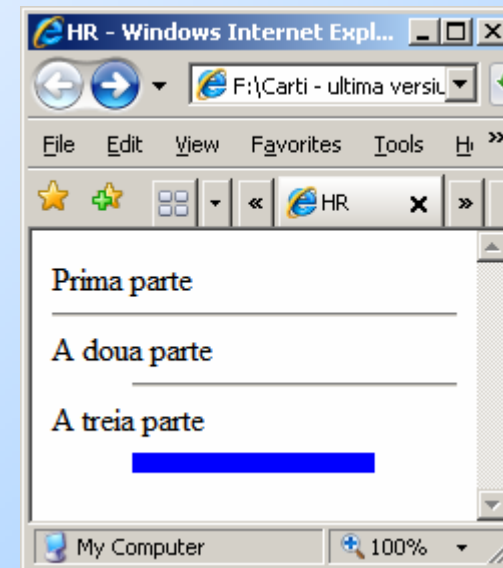
A doua parte

```
<HR WIDTH=60% ALIGN="right">
```

A treia parte

```
<HR WIDTH=60% size=10  
color="blue">
```

```
</BODY>
```



- ◆ Alinierea implicita este CENTER

Culori ancore

- ◆ Se pot seta culori pentru ancore (link-uri) functie de tipul lor: pagina, pagina vizitata, pagina activa – cea selectata (click)

```
<BODY BGCOLOR="#33FFFF" TEXT="#330033"  
LINK="#33CC00" VLINK="#FF00CC"  
ALINK="#009900">
```

```
<a href="test2.html">
```

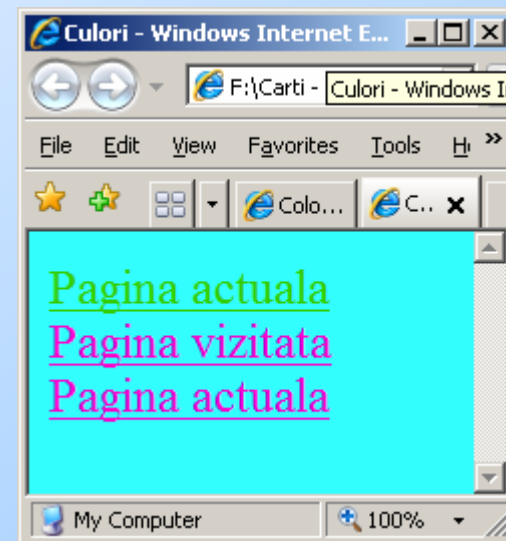
Pagina


```
<a href="test1.html">
```

Pagina vizitata


```
<a href="test.html">
```

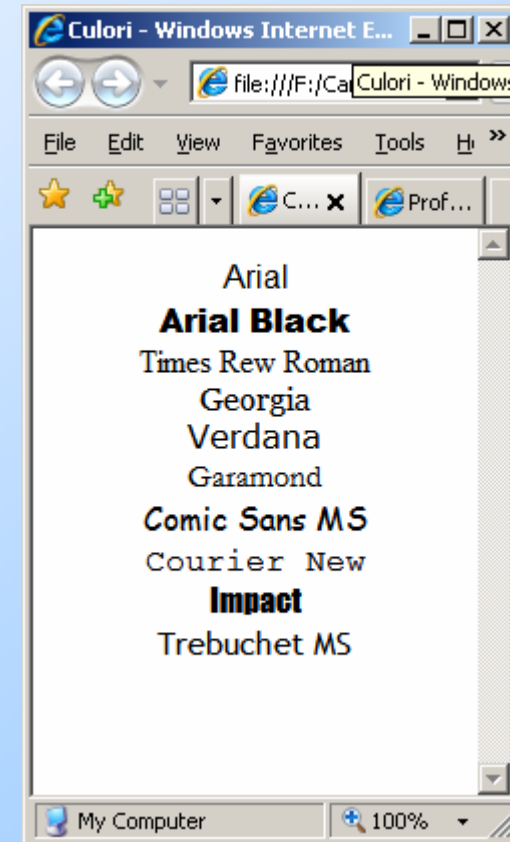
Pagina activa



Font - Face

◆ Se poate schimba fontul de scriere cu FONT FACE:

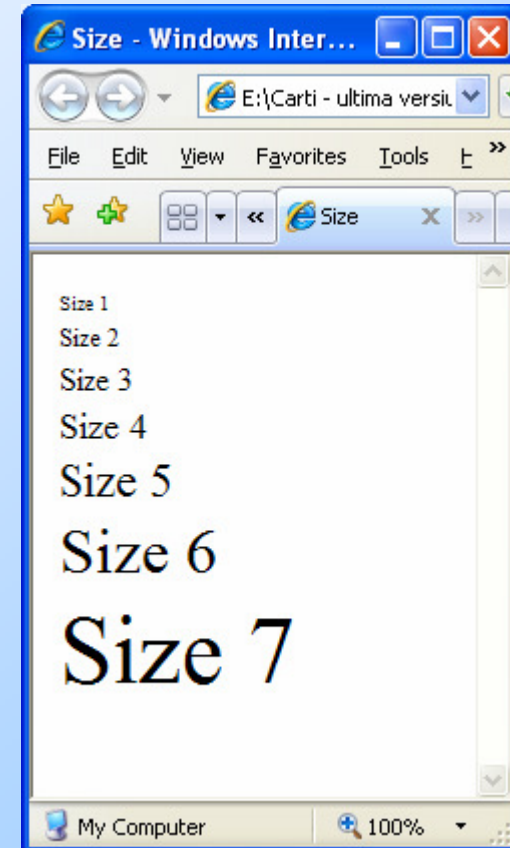
```
<body>
<P ALIGN="center">
<FONT FACE="Arial">Arial</FONT>
<br>
<FONT FACE="Arial Black">Arial Black</FONT>
<br>
<FONT FACE="Times Rew Roman">Times Rew Roman</FONT>
<br>
<FONT FACE="Georgia">Georgia</FONT>
<br>
<FONT FACE="Verdana">Verdana</FONT></body>
<br>
<FONT FACE="Garamond">Garamond</FONT>
<br>
<FONT FACE="Comic Sans MS">Comic Sans MS</FONT>
<br>
<FONT FACE="Courier New">Courier New</FONT>
<br>
<FONT FACE="Impact">Impact</FONT>
<br>
<FONT FACE="Trebuchet MS">Trebuchet MS</FONT>
</html>
```



Font Size

◆ Se poate dimensiona fontul cu FONT SIZE:

```
<TABLE BORDER=0><TR>  
<TD><FONT SIZE=1>Size 1</FONT></TD>  
</TR><TR>  
<TD><FONT SIZE=2>Size 2</FONT></TD>  
</TR><TR>  
<TD><FONT SIZE=3>Size 3</FONT></TD>  
</TR><TR>  
<TD><FONT SIZE=4>Size 4</FONT></TD>  
</TR><TR>  
<TD><FONT SIZE=5>Size 5</FONT></TD>  
</TR><TR>  
<TD><FONT SIZE=6>Size 6</FONT></TD>  
</TR><TR>  
<TD><FONT SIZE=7>Size 7</FONT></TD>  
</TR></TABLE>
```



Combinatii

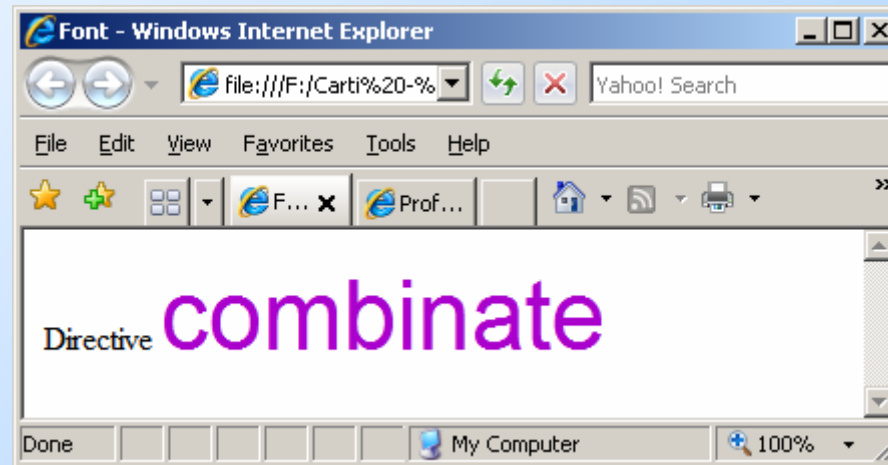
- ◆ Directivele anterioare se pot combina:

`<body>`

Directive `<FONT COLOR="#AA00CC"`

`FACE="ARIAL" SIZE="7">combinat`

`</BODY>`



Zona HEAD

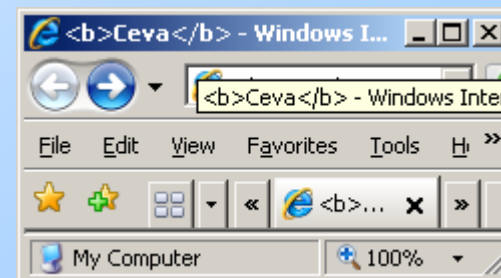
- ◆ In zona de header a paginii (intre `<head>` si `</head>`) putem pune o serie de directive de descriere a documentului.

HEAD - Title

- ◆ TITLE: titlul unei pagini. Sintaxa este:
`<title>titlu</title>`
- ◆ Titlul nu se formateaza, el nu apare in fereastra browserului.
- ◆ De exemplu:

`<title>Ceva</title>`

are efectul alaturat



HEAD - Base

- ◆ **Directiva <base>**: Spune calea de referință pentru paginile referite. Formatul său este:

```
<BASE HREF="adresa completa sau cale absoluta /  
relativa" >
```

- ◆ **Exemplu:**

```
<html>  
<head>  
<title>Fotografiile mele</title>  
<base href="http://www.un.server/imagini/eu">  
</head>  
<body>  
<h1>Acesta sunt eu</h1>  
  
  
  
  
</body>  
</html>
```

HEAD - Base

- ◆ În acest caz imaginile se găsesc în următoarele locații:

`http://www.un.server/imagini/eu/foto1.gif`

`http://www.un.server/imagini/eu/altele/foto2.gif`

`http://www.un.server/poze/foto3.gif`

`http://www.alt.server/imagini/foto4.gif`

- ◆ Directiva are efect asupra tuturor referințelor din pagina respectivă, inclusiv ancore, adrese ale scripturilor CGI apelate de forme, etc.

HEAD - META

- ◆ **Directiva <meta>**: Este folosită pentru a specifica "meta-informații despre pagina respectivă.
- ◆ Formatul acesteia este:

```
<META [HTTP-EQUIV | NAME]= "ceva"  
CONTENT= "altceva" >
```
- ◆ În cazul în care serverul face un inspectie al documentelor cerute se poate folosi *http-equiv* care are efectul de a instrui serverul să trimită un element de preambul al paginii definit de directiva respectivă.
- ◆ Iată un exemplu:

Exemplu

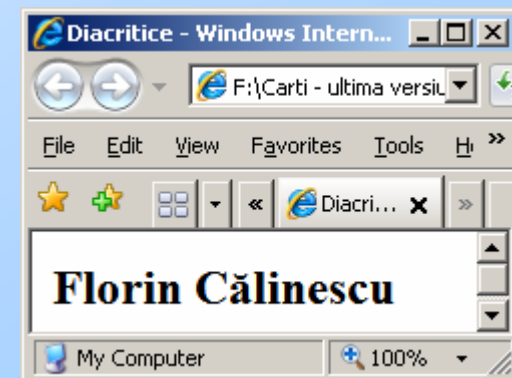
```
<html> <head>
<title> Programul conferintei </title>
<meta http-equiv="Expires"
  content="Sat, 30 Dec 2000 10:00:00 GMT">
</head>
<body>
<h1>Program</h1>
. . . . .
</body>
</html>
```

- ◆ Serverul va trimite un element de preambul de forma:

```
Expires: Sat, 30 Dec 2000 10:00:00 GMT
```

Alt exemplu

- ◆ Pentru a specifica setul de caractere folosit – de exemplu UTF-8 - se poate folosi:
- ◆ `<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
- ◆ In felul acesta putem diacritice in pagini web
- ◆ Documentul sursa trebuie salvat ca document UTF-8



Formulare

- ◆ Majoritatea directivelor HTML servesc pentru formatarea textului paginii. Formularele sunt cele care pot face ca o pagină să fie interactivă (să intre în dialog cu utilizatorul).
- ◆ Un formular trebuie să aibă în principiu:
 1. O acțiune asociată numită *script CGI*. Datele completate de utilizator sunt preluate de acest script și prelucrate. Rezultatul scriptului este trimis ca răspuns utilizatorului. Această acțiune se găsește în directiva *<form>*

Formulare

2. Unul sau mai multe câmpuri de tip text, buton, check-box, lista de alegere etc. Acestea sunt introduse prin directivele `<input>` din interiorul formularului.
3. Un buton prin selectarea căruia valorile completate sunt trimise serverului de web. Așa cum am văzut în primul capitol, el va lansa scriptul CGI și va trimite rezultatele către browser. Acest buton se implementează tot prin intermediul unei directive `<input>`.

Exemplu

- ◆ Forma generica a unui fișier HTML care conține un formular este următoarea:

```
<html>
<head>
<title>Titlu</title>
</head>
<body>
<form>
<input . . . >
<select> <option> . . . </select>
<textarea> </textarea>
. . . . .
<input type=submit>
</form>
</body> </html>
```

Exemplu

- ◆ `<form>` și `</form>` încadrează elementele constitutive ale formularului,
- ◆ `<input>`, `<select>` și `<textarea>` introduc câmpurile formularului
- ◆ prin directivele `<input type = submit>` se definesc butoane a căror selectare trimite datele completate în formă către serverul de web (pentru a fi transferate scriptului CGI care le prelucrează).

FORM

- ◆ Sintaxa acestei directive este următoarea:

```
<FORM ACTION="url" METHOD=[POST | GET]  
  [ENCTYPE="..." ]>
```

unde:

- ◆ *url* este adresa relativă sau absolută la care se găsește scriptul CGI asociat formularului,
- ◆ *method* arată modul în care scriptul CGI primește datele din formă: în cazul metodei *GET* acestea sunt primite într-o variabilă de mediu (environment) numită *QUERY_STRING* având lungimea *CONTENT_LENGTH* pe când în cazul metodei *POST* datele sunt primite la intrarea standard (*stdin*).

FORM

- ◆ Metoda POST este preferabilă atunci când volumul acestor date este mare.
- ◆ Dacă metoda nu este specificată, automat va fi considerată GET.
- ◆ *enctype* arată modul în care browserul codifică informația din formă pentru a o trimite serverului.
- ◆ Această opțiune este în general puțin folosită, în lipsă considerându-se tipul standard *application/x-www-form-urlencoded*.

INPUT

- ◆ **Directiva <input>**
- ◆ Sintaxa acestei directive este:

```
<INPUT TYPE=tip [NAME="nume" ]  
      [VALUE="valoare" ]  
      [SIZE=dimensiune_afisare ]  
      [MAXLENGTH=dimensiune_maxima  
      ] [SRC="url" ] [CHECKED]>
```

unde:

- ◆ TYPE = *tip* definește numele simbolului asociat câmpului.

INPUT

- ◆ NAME = "*nume*" definește numele simbolului asociat câmpului.
- ◆ VALUE = "*valoare*" definește valoarea inițială afișată de browser
- ◆ SIZE = *dimensiune_afisare* definește lungimea câmpului afișat de browser
- ◆ MAXLENGTH = *dimensiune_maxima* arată lungimea maximă a valorii care se poate tasta
- ◆ SRC = "*url*" definește o adresă relativă sau absolută (ca în cazul imaginilor - vezi tipul *img* mai jos)
- ◆ În funcție de tipul câmpului avem următoarele variante

Type=TEXT

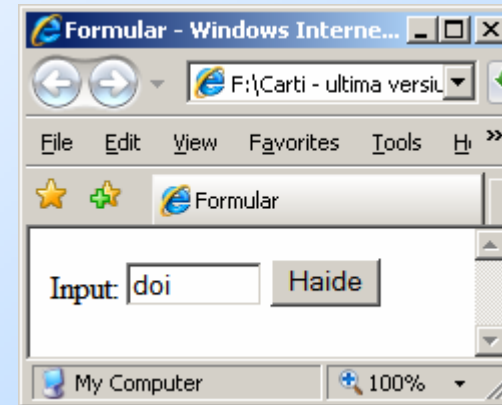
◆ Sintaxa:

```
<INPUT TYPE=TEXT NAME="nume"  
  [VALUE="valoare" ]  
  [MAXLENGTH=dimensiune_maxima ]  
  [SIZE=dimensiune_afisare ]>
```

- ◆ Definește un câmp de tip șir de caractere.
- ◆ Nu mutem tasta mai mult de MAXLENGTH caractere

Exemplu

```
<html>
<head>
<title>Formular</title>
</head>
<BODY>
<form action="unu.php">
Input: <input type="text" name="nume"
      value="doi" maxlength="5" size="7">
<input type="submit" value="Haide">
</form>
</BODY>
</html>
```



Type=SUBMIT

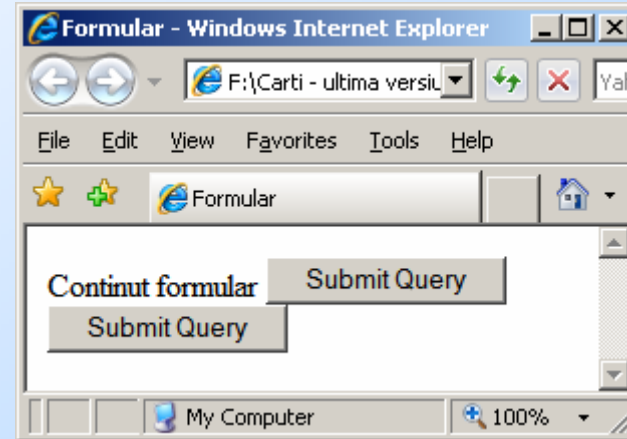
- ◆ Sintaxa:

```
<INPUT TYPE=SUBMIT [NAME="nume" ]  
  [VALUE="valoare" ]>
```

- ◆ Definește un buton prin selectarea căruia datele completate în formă sunt trimise serverului de web.
- ◆ Pot exista mai multe astfel de butoane, deosebite prin valoarea dată prin clauza *name*.
- ◆ Clauza *value* este folosită pentru a defini inscripționarea afișată de browser pentru acest buton (in lipsă inscripția este data de browser).

Exemplu

```
<html>
<head>
<title>Formular</title>
</head>
<BODY>
<form action="unu.php">
Continut formular
<input type="submit" name="buton1">
<input type="submit" name="buton2">
</form>
</BODY>
</html>
```



Type=RESET

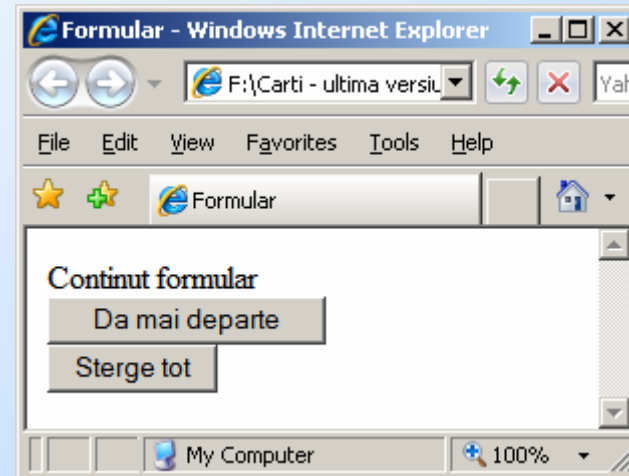
- ◆ Sintaxa:

```
<INPUT TYPE=RESET [VALUE="valoare" ]>
```

- ◆ Definește un buton prin a cărui selectare valorile câmpurilor formularului sunt aduse la starea inițială.
- ◆ Nu se face nici un schimb de date cu serverul (e interpretat de browser)
- ◆ Clauza *value* este folosită ca mai sus pentru inscripționarea lui.

Exemplu

```
<html>
<head>
<title>Formular</title>
</head>
<BODY>
<form action="unu.php">
Continut formular<br>
<input type="submit" value="Da mai departe">
<br>
<input type="reset" value="Sterge tot">
</form>
</BODY>
</html>
```



Type=PASSWORD

◆ Sintaxa:

```
<INPUT TYPE=PASSWORD NAME="nume"  
  [VALUE="valoare" ]  
  [MAXLENGTH="dimensiune_maxima" ]  
  [SIZE="dimensiune_afisare" ]>
```

- ◆ este asimilat tipului *text* cu deosebirea că ceea ce se tastează pe un astfel de câmp nu se afișează în clar pe ecran (în mod obișnuit se afișeaza în ecou caracterul asterisc - *) dar se transmite în clar serverului!

Exemplu

```
<html>
<head>
<title>Formular</title>
</head>
<BODY>
<form action="unu.php">
Dati parola: <input type="password"
               name="nume" value="doi" maxlength="5"
               size="7">
<input type="submit" value="Autentifica">
</form>
</BODY>
</html>
```



Type=IMAGE

- ◆ Sintaxa:
- ◆ `<INPUT TYPE=IMAGE NAME="nume" SRC="url"
[ALIGN=TOP|BOTTOM|MIDDLE] >`
- ◆ permite includerea unei imagini și selectarea unor anumite puncte ale sale.
- ◆ SRC arată adresa fișierului conținând imaginea iar *align* poziționarea acesteia.
- ◆ Nu este necesar buton de submit, click pe imagine declanșează transmiterea datelor (coordonate în imagine)

Type=HIDDEN

- ◆ Sintaxa:

```
<INPUT TYPE=HIDDEN NAME="nume"  
      VALUE="valoare" >
```
- ◆ Câmpurile de tip *hidden* nu apar pe ecran dar permit transmiterea de valori intr-o succesiune de pagini.
- ◆ Astfel de campuri apar de exemplu în formularele generate dinamic și sunt un mod prin care serverul setează încă de la transmiterea formularului către browser anumite perechi (*nume_simbol* – *valoare_simbol*) pentru acea formă.
- ◆ Un alt mod de a face acest lucru - in PHP si alte limbaje de scripting - este folosirea variabilelor de sesiune.

Type = CHECKBOX

- ◆ Sintaxa:
 - ◆ `<INPUT TYPE=CHECKBOX NAME="nume" VALUE="valoare" [CHECKED]>`
- ◆ Acest tip permite definirea unei casete care poate fi bifată sau nu (în genul unui comutator pornit/oprit).
- ◆ Dacă este prezentă clauza *CHECKED* caseta este inițial bifată, altfel nu.
- ◆ *Observație:* Spre deosebire de celelalte tipuri, doar pentru casetele bifate vor fi transmise de la browser către server perechi (*nume_simbol*, *valoare_simbol*). Dacă nici o casetă nu a fost bifată nici unul dintre simbolii asociați lor nu este prezent în șirul transmis.

Type = RADIO

- ◆ Sintaxa:

```
<INPUT TYPE=RADIO NAME="nume" VALUE="valoare"  
      [CHECKED]>
```

- ◆ Tipul *radio* este asemănător tipului *checkbox*, cu deosebirea ca numai un singur *buton radio* dintre cele cu același nume (dat de clauza *name*) poate fi selectat la un moment dat. Dacă este prezentă clauza CHECKED butonul respectiv este inițial selectat. Doar unul dintre butoanele radio cu același nume poate conține această clauză.

- ◆ *Observație:* În cazul tipului *checkbox* este uzual (dar nu necesar) ca mai multe casete prezente în formă să aibă aceeași valoare și nume diferite pe când în cazul *radio* butoanele au același nume și valori diferite, pentru un astfel de nume transmițându-se o singură pereche (*nume_simbol, valoare_simbol*) de la browser către server.

Liste SELECT

- ◆ Sintaxa acestei directive este:

```
<SELECT NAME="nume" [SIZE="dimensiune_fereastră"]>  
<OPTION [VALUE="valoare_1" ] [SELECTED]> text_afisat_1  
<OPTION [VALUE="valoare_2" ] [SELECTED]> text_afisat_2  
.  
.  
.  
</SELECT>
```

- ◆ Cu ajutorul acestei directive se definesc meniuri verticale din care se poate alege o singură valoare. Numele simbolului este cel din *SELECT* iar lista de valori este introdusă element cu element cu directiva *OPTION*. Dacă *VALUE* lipsește se consideră automat ca valoare textul corespunzător opțiunii respective (*text_afisat_i*).
- ◆ *Observație:* în fereastră (de dimensiune dată de *SIZE*) browserul afișează textele asociate iar selectarea unuia dintre ele asociază simbolului *nume* valoarea din clauza *VALUE* pentru textul respectiv.

TEXTAREA

- ◆ Sintaxa:

```
<TEXTAREA NAME="nume" [ROWS=numar_linii]
  [COLS=numar_coloane]
  [WRAP="soft" | "hard" | "off"]>
Text initial
</TEXTAREA>
```

- ◆ Aceasta directivă este asemănătoare cu `<input type=text>` cu deosebirea că browserul va permite tastarea unui text într-o fereastră cu mai multe linii (*nr_linii*). Dimensiunea pe orizontală a ferestrei este dată de *numar_coloane*.

TEXTAREA

- ◆ Clauza *WRAP* (nerecunoscută în unele cazuri) arată modul în care browserul afișează textul în fereastra respectivă și cum îl transmite serverului:
- ◆ în cazul *soft* tastarea unui text mai mare decât *nr_coloane* duce la afișarea pe mai multe linii dar la transmitere textul e liniarizat
- ◆ în cazul *hard* afișarea se face ca mai sus dar transmiterea se face sub forma mai multor linii, așa cum apar în fereastră.
- ◆ În cazul *off* (care este implicit) textul e afișat și transmis așa cum este tastat.

Exemplu

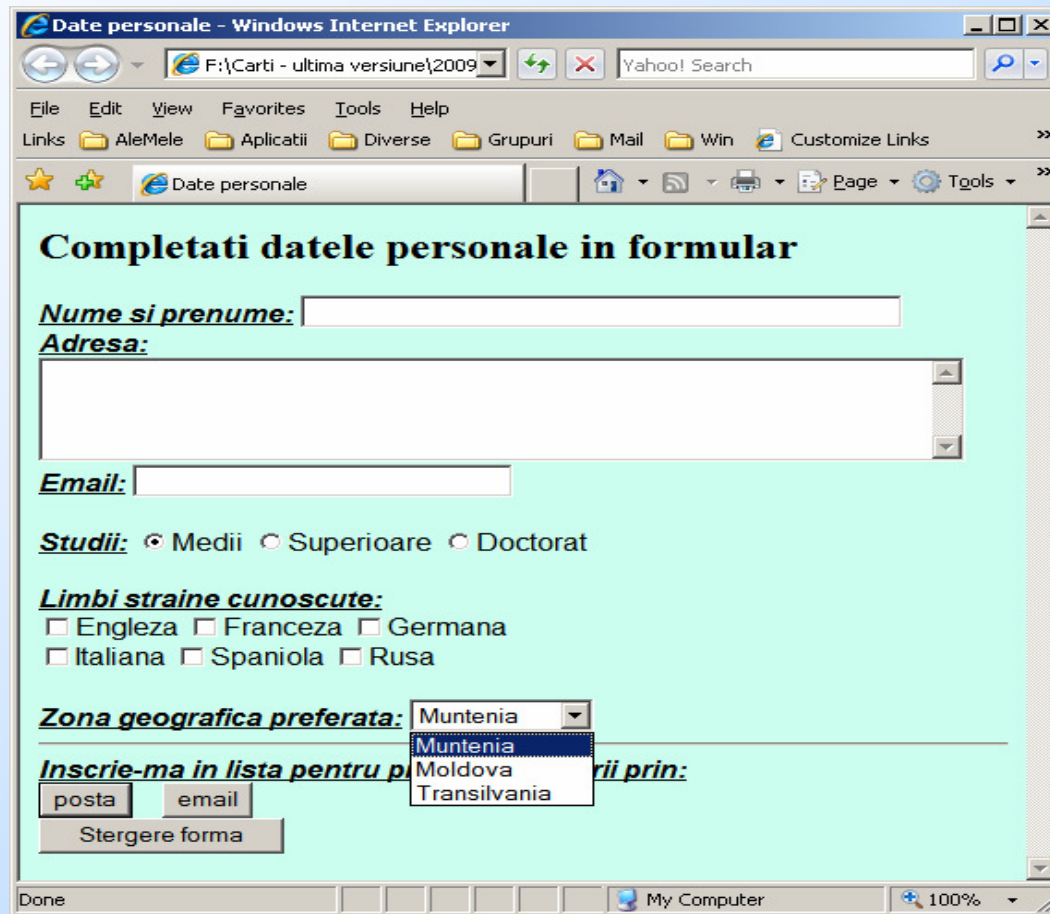
```
<html><head><title>Date personale</title></head>
<body bgcolor="CCFFF">
<h2>Completati datele personale in formular</h2>
<font size = 3 face = "arial">
<form action="inregistrare.cgi" method=POST>
<u><b><i>Nume si prenume:</i></b></u> <input name="nume"
    size=50>
<br>
<u><b><i>Adresa:</i></b></u><br>
    <textarea name="adresa" rows=4 cols=60 wrap="soft">
    </textarea>
<u><b><i>Email:</i></b></u> <input type=text name="email"
    size=30>
<p>
<u><b><i>Studii:</i></b></u>
<input type=radio name="studii" value="medii" checked>Medii
<input type=radio name="studii" value="superioare">Superioare
<input type=radio name="studii" value="doctorat">Doctorat
```

Exemplu

```
<p>
<u><b><i>Limbi straine cunoscute:</i></b></u> <br>
<input type=checkbox name="limba" value="engleza">Engleza
<input type=checkbox name="limba" value="franceza">Franceza
<input type=checkbox name="limba" value="germana">Germana
<br>
<input type=checkbox name="limba" value="italiana">Italiana
<input type=checkbox name="limba" value="spaniola">Spaniola
<input type=checkbox name="limba" value="rusa">Rusa
<p>
<u><b><i>Zona geografica preferata:</i></b></u>
<select name="zona">
<option value="muntenia">Muntenia
<option value="moldova">Moldova
<option value="transilvania">Transilvania
</select>
<hr>
<u><b><i>Inscrie-ma in lista pentru primirea brosurii
    prin:</i></b></u><br>
<input type=submit name="tip" value="posta">&nbsp; &nbsp;
<input type=submit name="tip" value="email"><br>
<input type=reset value="Stergere forma">
</form>
</body> </html>
```

Rezultat

◆ Rezultatul afisat pe ecran:



The screenshot shows a Windows Internet Explorer browser window titled "Date personale - Windows Internet Explorer". The address bar displays "F:\Carti - ultima versiune\2009" and the search bar contains "Yahoo! Search". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The toolbar shows various icons for navigation and actions. The main content area displays a form titled "Completati datele personale in formular". The form fields include:

- Nume si prenume:** A text input field.
- Adresa:** A large text area for address input.
- Email:** A text input field.
- Studii:** Radio buttons for "Medii", "Superioare", and "Doctorat".
- Limbi straine cunoscute:** Checkboxes for "Engleza", "Franceza", "Germana", "Italiana", "Spaniola", and "Rusa".
- Zona geografica preferata:** A dropdown menu currently showing "Muntenia" with a list of options: "Muntenia", "Moldova", and "Transilvania".
- Inscrie-ma in lista pentru primirea prin:** Checkboxes for "posta" and "email".
- Stergere forma:** A button to clear the form.

The status bar at the bottom shows "Done", "My Computer", and "100%" zoom level.

TABELE

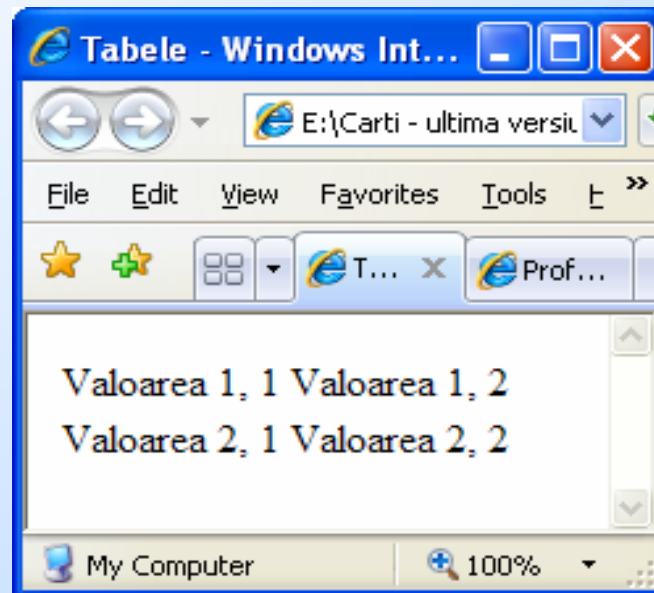
- ◆ Cele 3 directive (taguri) de baza pentru lucrul cu tabele sunt:
 1. `<table>` - defineste inceputul unei tabele
 2. `<tr>` - inceputul unei linii intr-o tabela
 3. `<td>` - o celula pe o linie dintr-o tabela.
- ◆ Fiecare dintre cele 3 directive anterioare se incheie cu `</directiva>`

Exemplu

```
<table>
<tr><!-- Linia 1 -->
<td><!-- Prima celula -->
Valoarea 1, 1
</td>
<td><!-- A doua celula -->
Valoarea 1, 2
</td>
</tr>
<tr><!-- Linia 2 -->
<td><!-- Prima celula -->
Valoarea 2, 1
</td>
<td><!-- A doua celula -->
Valoarea 2, 2
</td>
</tr>
</table>
```

REZULTAT

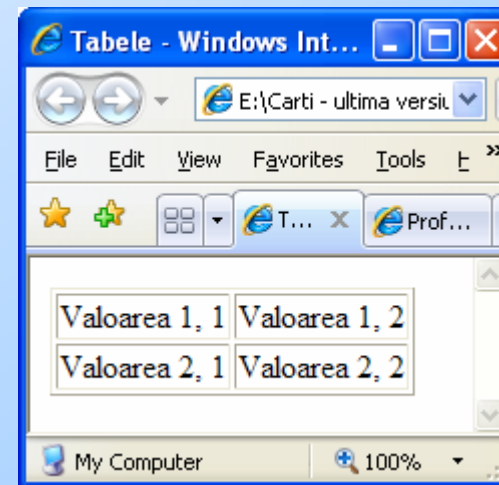
- ◆ Ca rezultat browserul va afisa:



BORDER

- ◆ Din cate se vede in exemplul anterior, implicit caroiajul un este vizibil.
- ◆ Pentru a-l vizualiza directiva table poate primi parametrul BORDER
- ◆ Exemplu:

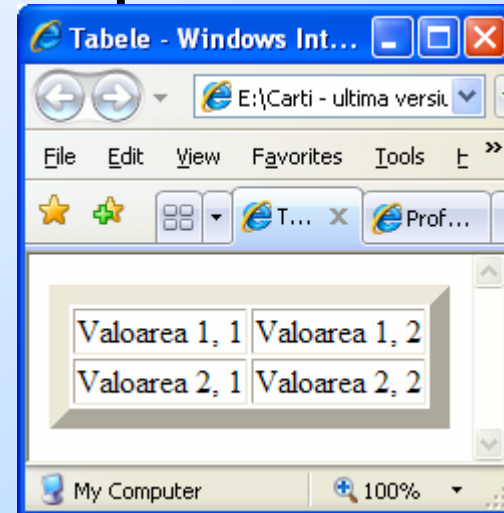
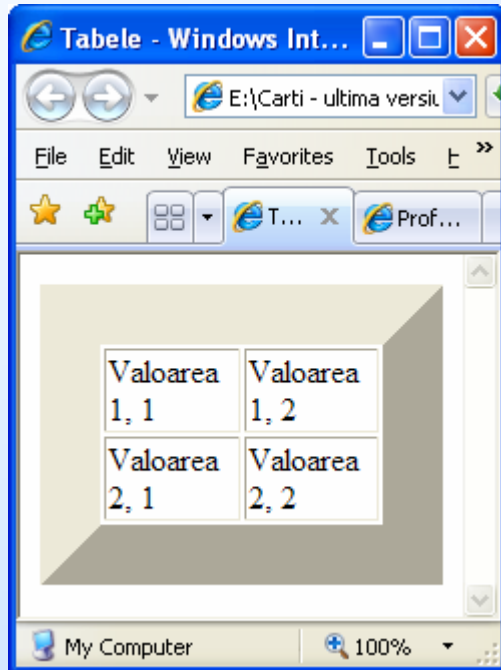
```
<table border=1>
```



BORDER

- ◆ Valoarea parametrului poate fi mai mare ca 1:

`<table border=10>`



`<table border=30>`

WIDTH

- ◆ Un alt parametru la border este WIDTH.
- ◆ Exprimat in procente parametrul arata cat din dimensiunea disponibila ocupa tabela
- ◆ Exemplu: 50%, 75%, 100%:

```
<table border=1 width=50%>
```

```
. . .
```

```
</table>
```

```
<table border=1 width=70%>
```

```
. . .
```

```
</table>
```

```
<table border=1 width=100%>
```

```
. . .
```

```
</table>
```

WIDTH

◆ Rezultat:

Valoarea 1, 1	Valoarea 1, 2
Valoarea 2, 1	Valoarea 2, 2
Valoarea 1, 1	Valoarea 1, 2
Valoarea 2, 1	Valoarea 2, 2
Valoarea 1, 1	Valoarea 1, 2
Valoarea 2, 1	Valoarea 2, 2

HEIGHT

- ◆ Similar cu WIDTH, dar pe verticala.
- ◆ Se poate exprima in cifre absolute sau in procente
- ◆ Exemplu:

```
<table border=1 width=50% height=50>
```

```
. . .
```

```
</table>
```

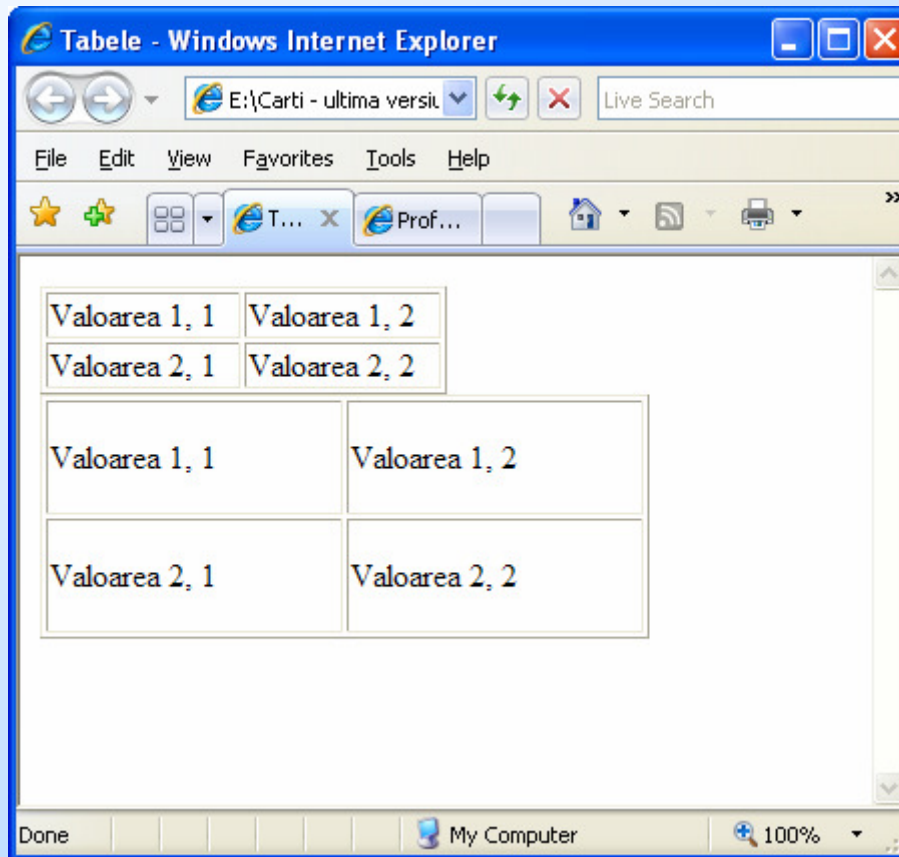
```
<table border=1 width=70% height=50%>
```

```
. . .
```

```
</table>
```

Rezultat

◆ Obținem:



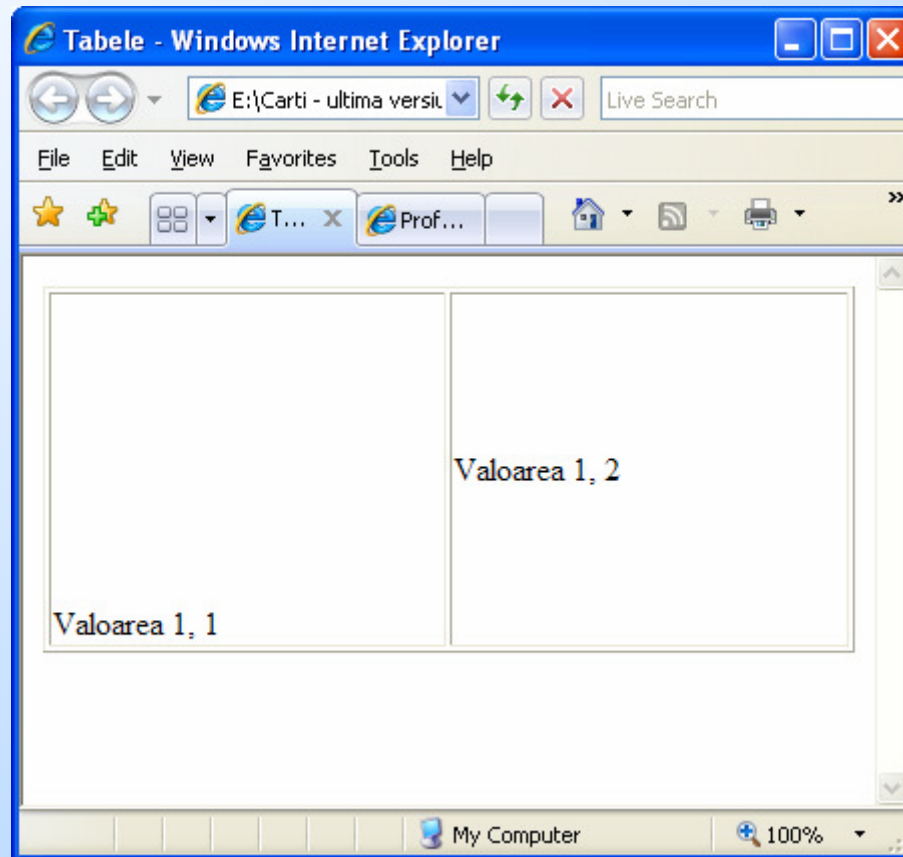
ALIGN

- ◆ Pozitionarea valorilor in celule se face formatandu-le individual.
- ◆ Exemplu:

```
<table border=1 width=100 height=75>  
<tr><!-- Linia 1 -->  
<TD ALIGN="left" VALIGN="bottom">  
<!-- Prima celula -->  
Valoarea 1, 1  
</td>  
<td><!-- A doua celula -->  
Valoarea 1, 2  
</td>  
</tr>  
</table>
```

Rezultat

◆ Obtinem:



Aliniere

- ◆ Pentru alinierea pe orizontala putem folosi:
 - ◆ LEFT,
 - ◆ RIGHT,
 - ◆ CENTER
- ◆ Pentru alinierea pe verticala putem folosi:
 - ◆ TOP,
 - ◆ BOTTOM,
 - ◆ MIDDLE

Continut celule

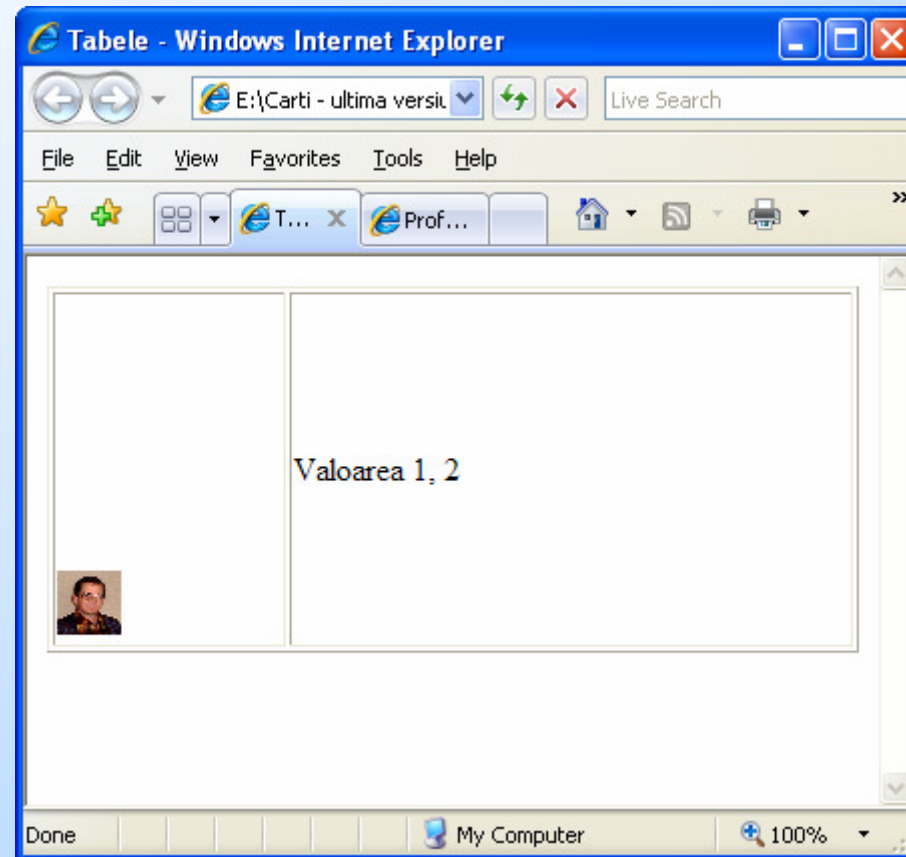
- ◆ Celulele unei tabele pot contine un numai text ci orice altcela, inclusiv imagini.

- ◆ Exemplu:

```
<table border=1 width=100% height=75%>
<tr><!-- Linia 1 -->
<TD ALIGN="left" VALIGN="bottom"><!-- Prima
    celula -->
<IMG SRC="foto2.jpg" WIDTH=32 HEIGHT=32 ALT="Ed">
</td>
<td><!-- A doua celula -->
Valoarea 1, 2
</td>
</tr>
</table>
```

Rezultat

◆ Obtinem:



Dimensiune celule

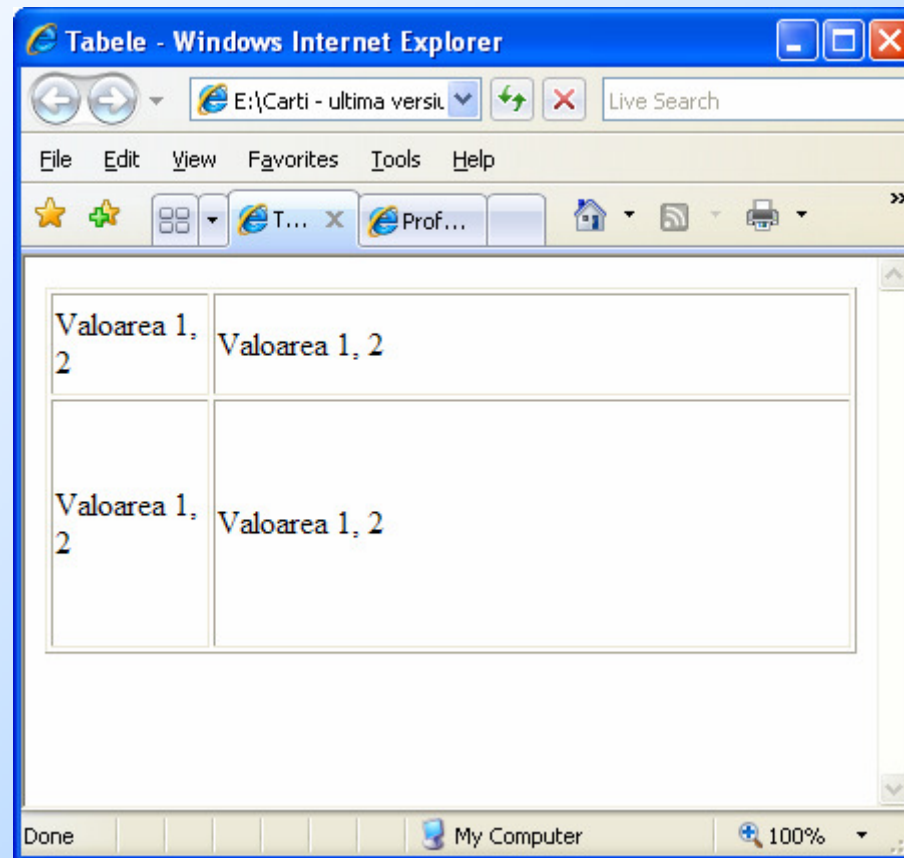
- ◆ WIDTH si HEIGHT, in cifre absolute sau procente, se pot pune ca parametri si la directiva TD
- ◆ In acest fel putem sa avem dimensiuni diferite pentru coloanele si liniile unei tabele.
- ◆ Parametrii de la TABLE arata dimensiunile tabelei in fereastra, cele de la TD pe cele ale celulei in tabela.

Exemplu

```
<table border=1 width=100% height=75%>
<tr><!-- Linia 1 -->
<td width=20% height=25%><!-- Prima celula -->
Valoarea 1, 2
</td>
<td width=80%><!-- A doua celula -->
Valoarea 1, 2
</td>
</tr>
<tr><!-- Linia 2 -->
<td height=60%><!-- Prima celula -->
Valoarea 1, 2
</td>
<td><!-- A doua celula -->
Valoarea 1, 2
</td>
</tr>
```

Rezultat

◆ Obtinem:



The screenshot shows a Windows Internet Explorer browser window titled "Tabele - Windows Internet Explorer". The address bar displays "E:\Carti - ultima versi...". The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help), a toolbar with navigation and utility icons, and a status bar at the bottom showing "Done", "My Computer", and "100%". The main content area displays a table with two rows and two columns. Each cell in the table contains the text "Valoarea 1, 2".

Valoarea 1, 2	Valoarea 1, 2
Valoarea 1, 2	Valoarea 1, 2

Celule goale

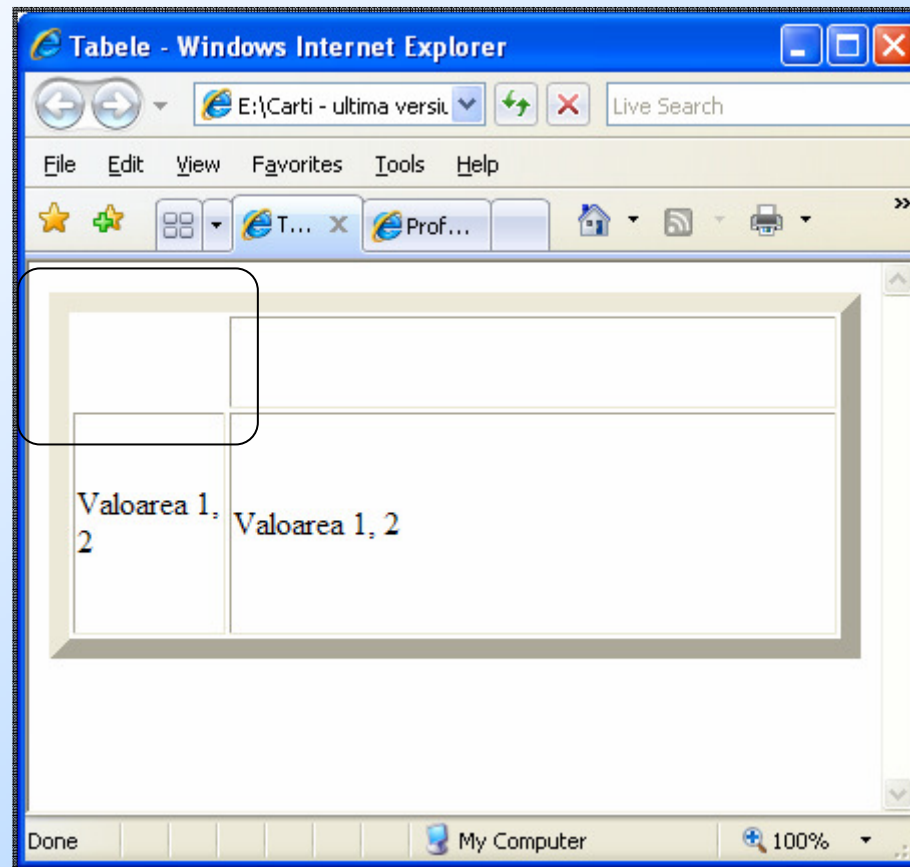
- ◆ Daca o celula este goala, browserul o arata deosebit de cele care contin ceva, inclusiv de cele care contin un spatiu de tip
- ◆ Exemplu:

. . . .

```
<tr><!-- Linia 1 -->  
<td width=20% height=25%><!-- nimic --></td>  
<td width=80%><!-- A doua celula -->&nbsp;</td>  
</tr> . . . .
```

Rezultat

◆ Obtinem:



CELLPADDING & CELLSPACING

- ◆ Se pun la TABLE
- ◆ CELLPADDING Arata cat spatiu este intre liniile caroiajului si continutul celulei.
- ◆ Implicit este luata o valoare (1 de exemplu)
- ◆ CELSPACING arata spatiul intre celule.
- ◆ Implicit este luata o valoare (2 de exemplu)

Exemplu

Tabela1

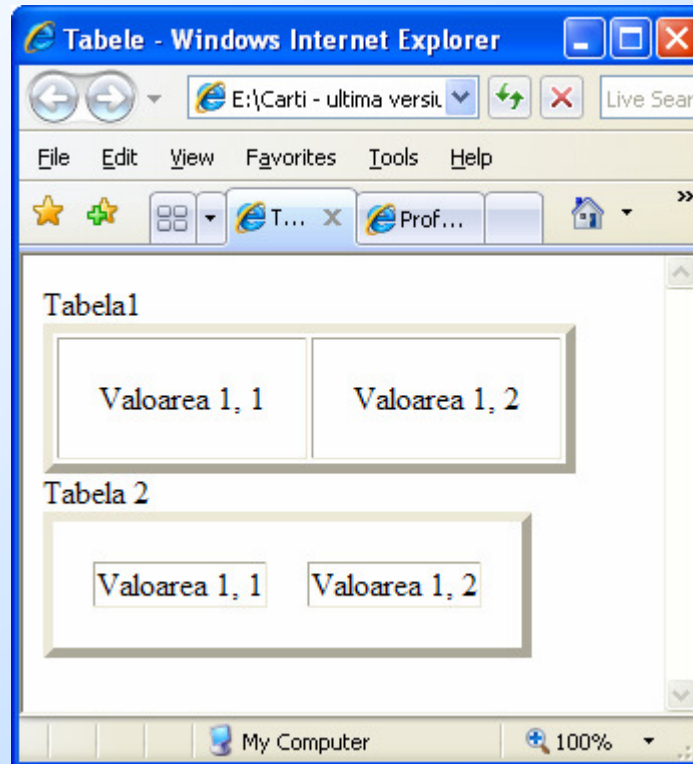
```
<table border=5 cellpadding=20>
<tr><!-- Linia 1 --><td><!-- Prima celula -->
Valoarea 1, 1
</td><td><!-- A doua celula -->
Valoarea 1, 2
</td></tr>
</table>
```

Tabela 2

```
<table border=5 cellspacing=20>
<tr><!-- Linia 1 --><td><!-- Prima celula -->
Valoarea 1, 1
</td><td><!-- A doua celula -->
Valoarea 1, 2
</td></tr>
</table>
```

Rezultat

◆ Obtinem:



◆ Cele 2 se pot bineinteles combina.

Sa coloram putin

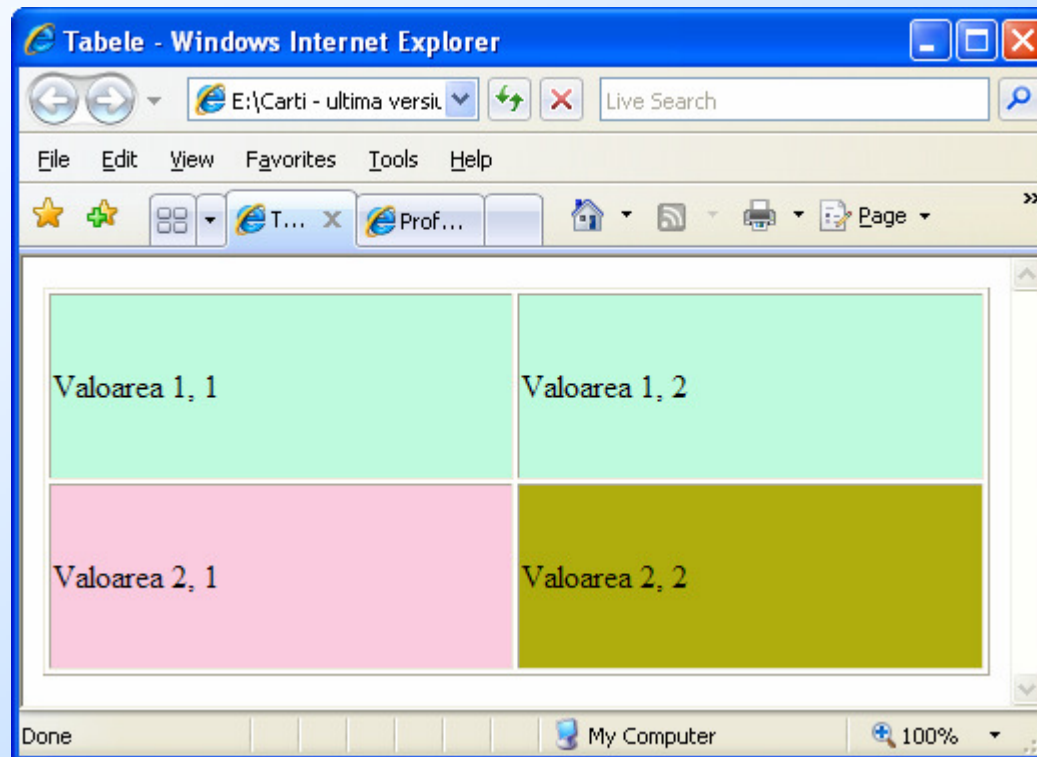
- ◆ Celulele unei tabele se pot colora (culoare fundal celula).
- ◆ Putem coloraa toata tabela, doar o linie sau doar o celula.
- ◆ Pentru asta se pune parametrul BGCOLOR la TABLE, TR respectiv la TD

Exemplu

```
<table border=1 width=100% height=100%>
<tr BGCOLOR=#BEFADE><!-- Linia 1 -->
<td><!-- Prima celula -->Valoarea 1, 1</td>
<td><!-- A doua celula -->Valoarea 1, 2</td>
</tr>
<tr><!-- Linia 2 -->
<td BGCOLOR=#FACADE><!-- Prima celula -->Valoarea
  2, 1</td>
<td BGCOLOR=#AFAD0E><!-- A doua celula --
  >Valoarea 2, 2</td>
</tr>
</table>
```

Rezultat

◆ Obtinem:



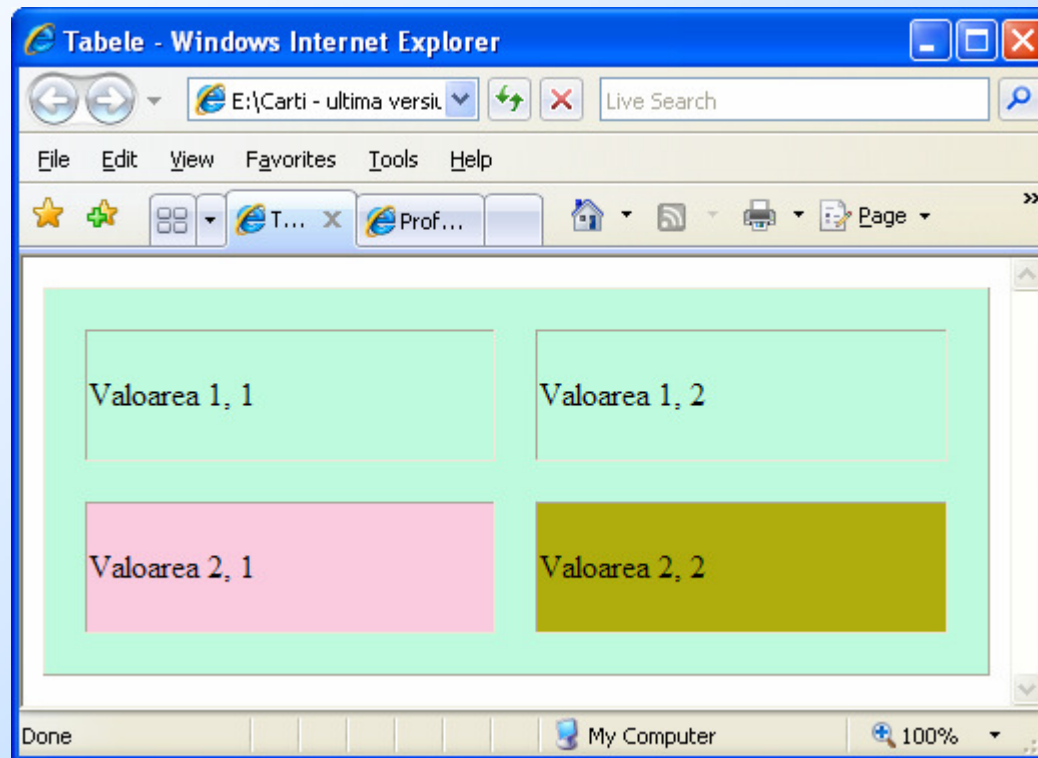
Alta colorare

- ◆ Daca mutam parametrul de culoare la tabela, va fi colorat inclusiv spatiul dintre celule:

```
<table border=1 cellspacing=20 width=100%
  height=100% BGCOLOR=#BEFADE>
<tr><!-- Linia 1 -->
<td><!-- Prima celula -->Valoarea 1, 1</td>
<td><!-- A doua celula -->Valoarea 1, 2</td>
</tr>
<tr><!-- Linia 2 -->
<td BGCOLOR=#FACADE><!-- Prima celula --
  >Valoarea 2, 1</td>
<td BGCOLOR=#AFAD0E><!-- A doua celula --
  >Valoarea 2, 2</td>
</tr>
</table>
```

Rezultat

◆ Obtinem:



Prioritate

- ◆ Prioritate are culoarea de celula, daca exista
- ◆ Daca un se coloreaza cu culoarea de linie, daca exista.
- ◆ Daca un se foloseste culoarea de tabela (daca exista).

COLSPAN si ROWSPAN

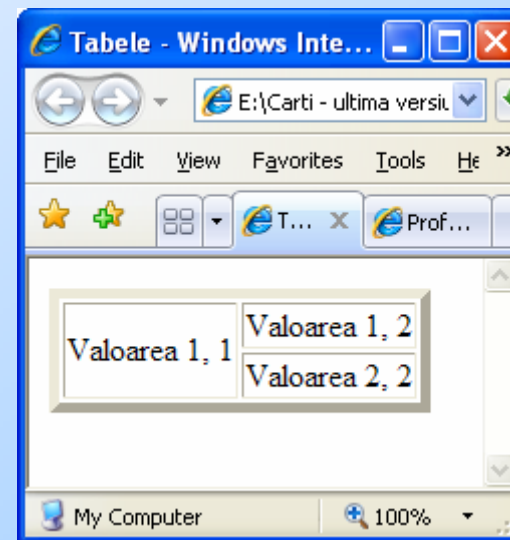
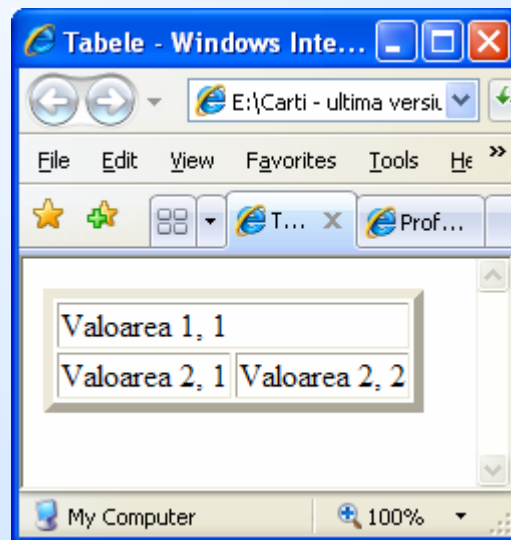
- ◆ Daca vrem ca o celula se ocupe mai multe coloane sau mai multe linii, la TD folosim parametrii COLSPAN respectiv ROWSPAN.
- ◆ Valoarea lor spune cate coloane, respectiv linii ocupa acea celula.

Exemplu

```
◆ <table border=5>
◆ <tr><!-- Linia 1 -->
◆ <td colspan=2><!-- Prima celula -->
◆ Valoarea 1, 1
◆ </td>
◆ </tr>
◆ <tr><!-- Linia 2 -->
◆ <td><!-- Prima celula -->
◆ Valoarea 2, 1
◆ </td>
◆ <td><!-- A doua celula -->
◆ Valoarea 2, 2
◆ </td>
◆ </tr>
◆ </table>
```

Rezultat

- ◆ Prima imagine este pentru colspan=2, a doua pentru o varianta similara cu rowspan=2



Formatare text celula

- ◆ Intr-o celula poate fi un intreg document HTML, cu toate elementele de formatare posibile.
- ◆ Pot exista imagini, ancore, directive de formatare text, etc

Exemplu

```
<table border=5 width=100% height=100%>
<tr><!-- Linia 1 -->
<td rowspan=2><!-- Prima celula -->
<h1>Intrebarea 1</h1>
<font size=4>Cine a spus "Invatati, invatati, invatati!"?
<ol>
<li>Tatal meu cand imi vedea notele
<li>Dirigintele, saptamanal
<li>Indrumatorul de an, la facultate
</ol>
Raspunsul <a href="www.google.ro">il gasiti pe aici</a>
</td>
<td>Valoarea 1, 2</td></tr>
<tr><!-- Linia 2 --><td><!-- A doua celula -->Valoarea 2,
    2</td>
</tr></table>
```

Rezultat

◆ Obtinem:

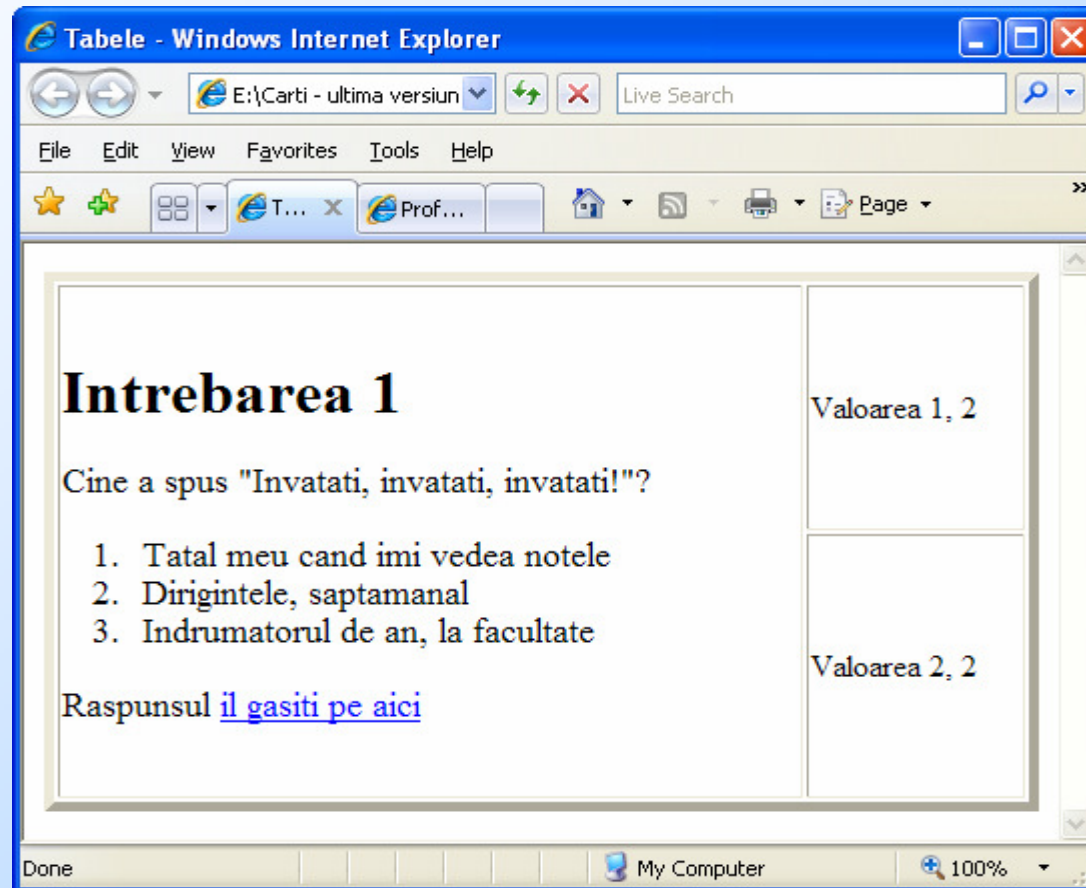
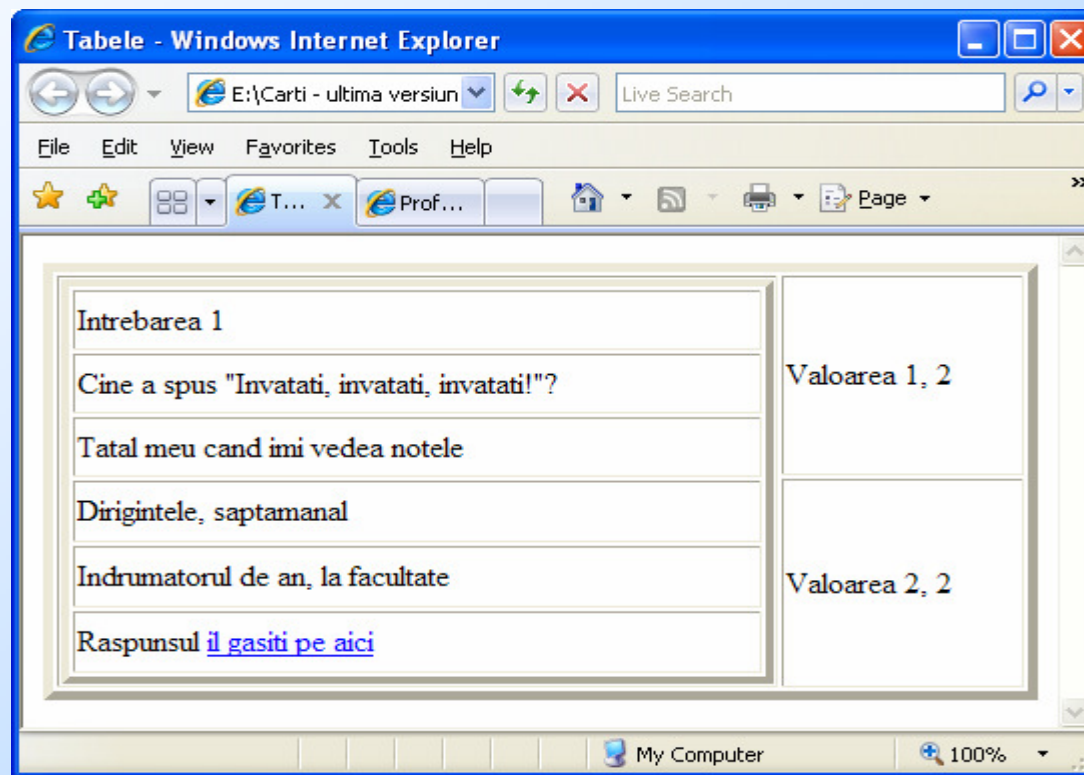


Tabela in tabela

- ◆ Intr-o celula a unei tabele se poate pune inclusiv o alta tabela:



Codul HTML pentru asta

```
<table border=5 width=100% height=100%>
<tr><!-- Linia 1 -->
<td rowspan=2><!-- Prima celula -->
<table border=4 width=100% height=100%
<tr><td>Intrebarea 1</td></tr>
<tr><td>Cine a spus "Invatati, invatati, invatati!"?</td></tr>
<tr><td>Tatal meu cand imi vedea notele</td></tr>
<tr><td>Dirigintele, saptamanal</td></tr>
<tr><td>Indrumatorul de an, la facultate</td></tr>
<tr><td>Raspunsul <a href="www.google.ro">il gasiti pe
    aici</a></td></tr>
</table>
</td>
<td>Valoarea 1, 2</td></tr>
<tr><!-- Linia 2 --><td><!-- A doua celula -->Valoarea 2,
    2</td>
</tr></table>
```


CADRE (Frames)

- ◆ Cu ajutorul lor putem imparti fereastra browserului in mai multe ferestre distincte.
- ◆ Iata un exemplu in care avem 2 ferestre. Avem nevoie de 3 pagini HTML, una master si doua cu continutul celor 2 ferestre.

Exemplu

- ◆ Pagina principala (atentie, lipseste directiva `<BODY>!`):

- ◆ `<html>`

- ◆ `<head>`

- ◆ `<title>Cadre</title>`

- ◆ `</head>`

- ◆ `<FRAMESET COLS="40%, *">`

- ◆ `<FRAME SRC="stanga.html">`

- ◆ `<FRAME SRC="dreapta.html">`

- ◆ `</FRAMESET>`

- ◆ `</html>`

Exemplu

◆ Cadrul stanga.html

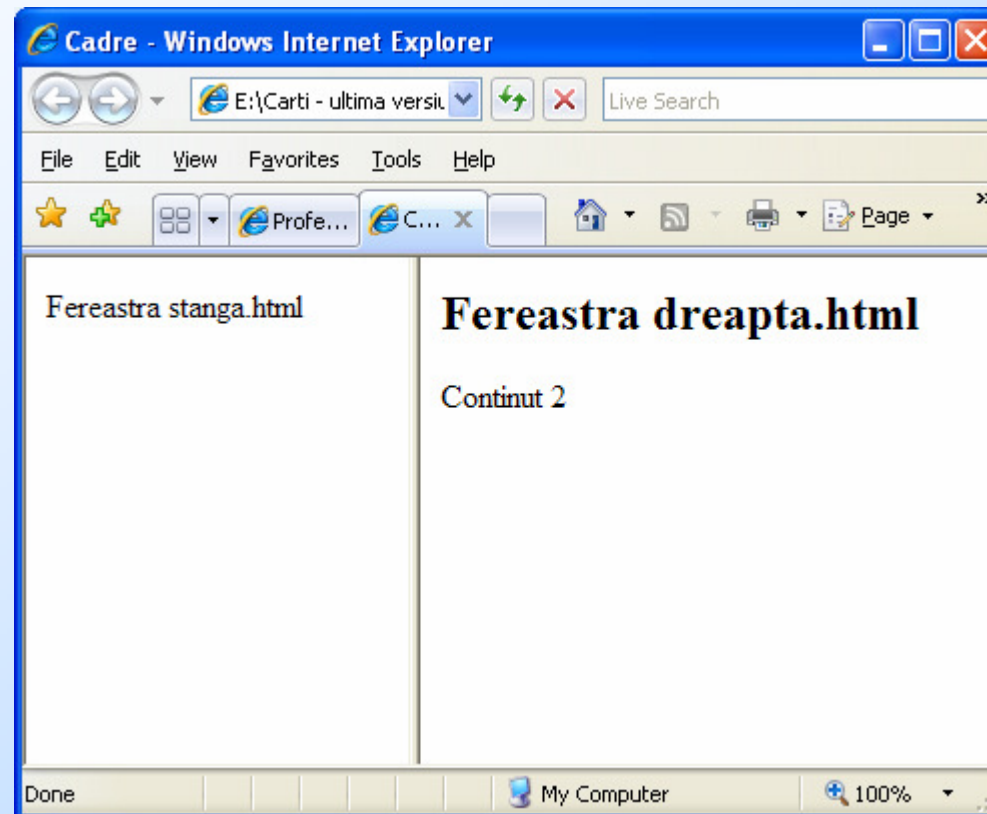
```
<html>  
<body>  
Fereastră stanga.html  
</body>  
</html>
```

◆ Cadrul dreapta.html:

```
<html>  
<body>  
<h2>Fereastră dreapta.html</h2>  
Continut 2  
</body>  
</html>
```

Rezultat

◆ Obtinem:

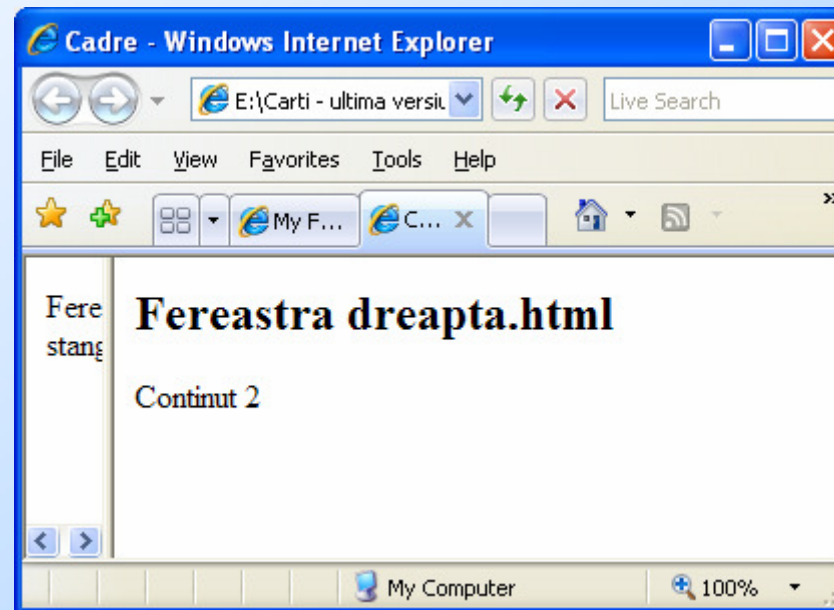


Dimensiuni

- ◆ Dimensiunea unui cadru se poate da:
- ◆ Procentual (din marimea ferestrei, => redimensionare o data cu fereastra)
- ◆ Absolut (in pixeli). De exemplu, daca in cazul anterior in loc de 40% foloseam 40 rezultatul era o dimensionare in pixeli, fara redimensionari o data cu fereastra:

Rezultat

◆ Obtinem:



Observatii

- ◆ Daca folosim valori absolute trebuie ca in lista de la FRAMESET unul dintre cadre sa fie dat dimensionat cu * (pentru redimensionarea o data cu fereastra).

- ◆ Se poate folosi si constructia:

<FRAMESET COLS="50, *, 2*">

care inseamna: cadrul 1 50 pixeli, restul pentru cadrele 2 si 3 dar cadrul 3 de doua ori mai mare decat cadrul 2.

Cadre imbricate

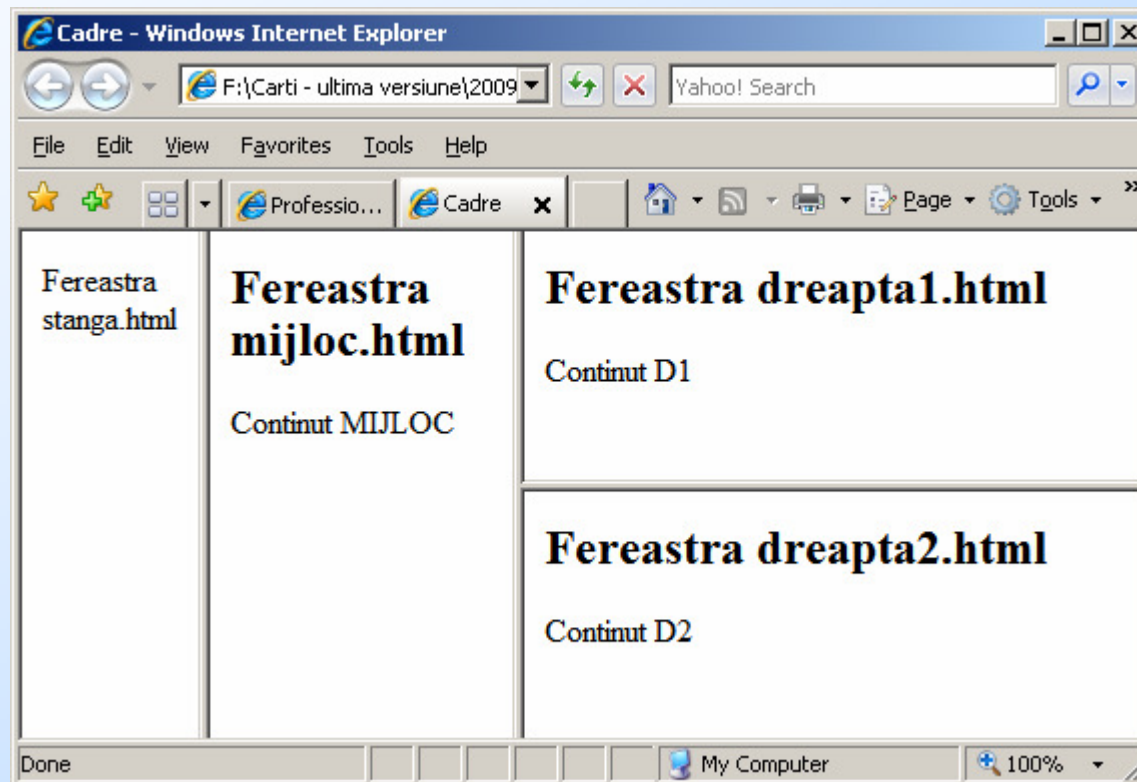
◆ Cadrele se pot si imbrica.

◆ Exemplu:

```
<html>
<head>
<title>Cadre</title>
</head>
<FRAMESET COLS="90,*,2*">
  <FRAME SRC="stanga.html">
  <FRAME SRC="mijloc.html">
  <FRAMESET ROWS="50%,50%">
    <FRAME SRC="dreapta1.html">
    <FRAME SRC="dreapta2.html">
  </FRAMESET>
</FRAMESET>
</html>
```


Rezultat

◆ Obtinem:



BORDER

- ◆ Zona dintre cadre se poate dimensiona si colora.
- ◆ Daca in exemplul anterior linia este:

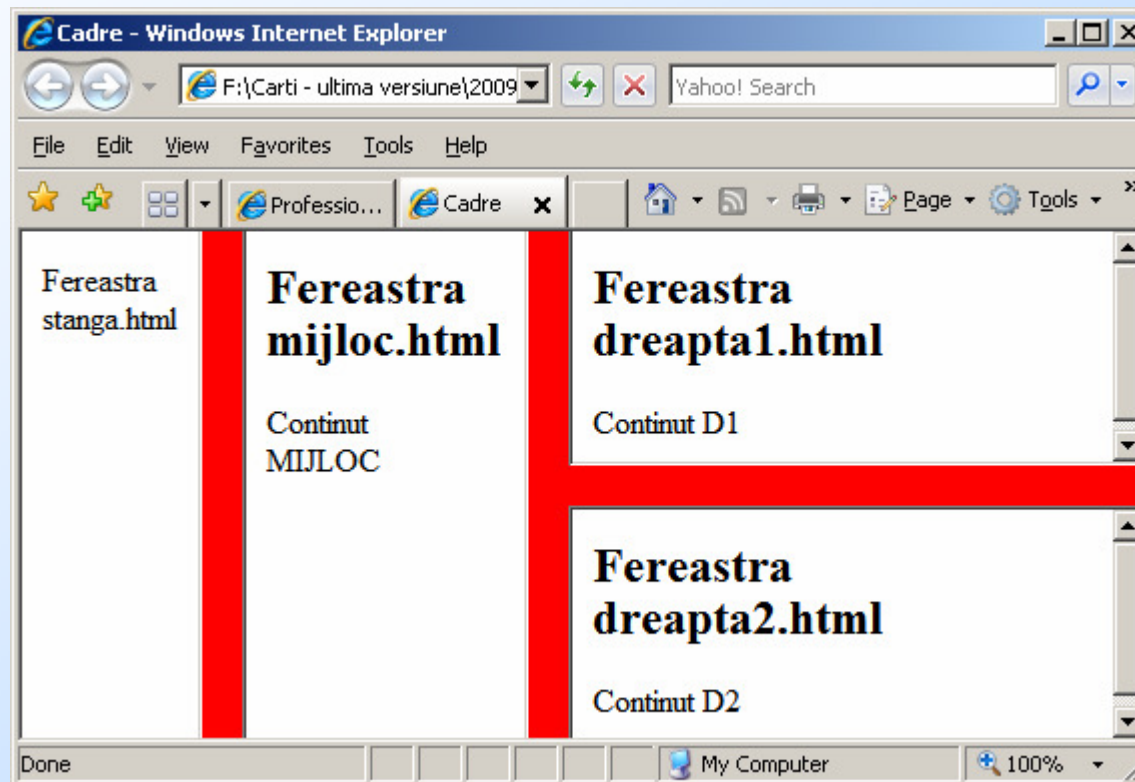
```
<FRAMESET COLS="90, *, 2*" BORDER=20  
  BORDERCOLOR="#FF0000">
```

obtinem rezultatul din transparentul urmator.

- ◆ BORDER da grosimea benzii iar BORDERCOLOR culoarea.

REZULTAT

◆ Obtinem:



Legaturi intre cadre

- ◆ Se poate ca un link dintr-un cadru sa deschida o pagina in alt cadru.
- ◆ Pentru aceasta, la FRAME trebuie sa dam parametrul NAME pentru a-i aloca un nume iar la <A HREF> parametrul TARGET pentru a indica fereastra in care se deschide noua pagina

Exemplu

- ◆ Pagina principala (avem doar doua cadre):

```
<FRAMESET COLS="33%,67%">
```

```
<FRAME SRC="stanga.html">
```

```
<FRAME SRC="dreapta.html"  
  NAME="FDREAPTA">
```

```
</FRAMESET>
```

Exemplu

- ◆ In pagina din stanga putem avea atunci un link de forma:

```
<A HREF="doc.html" TARGET =  
  "FDREAPTA"> Vezi doc.</A>
```

- ◆ La activarea acestui link pagina doc.html se va deschide in cadrul din dreapta

TARGET

- ◆ La target se mai pot folosi: si destinatiile predefinite `_top`, `_blank`, `_self` si `_parent`
- ◆ `_TOP` deschide pagina in fereastra intreaga de browser (fara cadre)
- ◆ `_BLANK` deschide pagina intr-o fereastra noua de browser (nu cadru!)
- ◆ `_SELF` deschide pagina in aceeaasi fereastra de browser sau in acelasi cadru (deci poate lipsi, e actiunea implicita)
- ◆ `_PARENT` deschide pagina in fereastra tata a cadrului curent (cea in care este definit prin `FRAMESET` cadrul curent, un acelasi lucru cu `_TOP`!)

ROWS

- ◆ Asa cum COLS imparte fereastra curenta (sau cadrul) pe verticala, ROWS face o impartire pe orizontala.
- ◆ Parametrii descrisi la COLS se pot folosi si la ROWS
- ◆ Intr-un exemplu anterior al doilea frameset e pe orizontala

Alte elemente

- ◆ Fiecare cadru poate avea o alta culoare de background sau imagine (se comporta ca o fereastră obisnuita de browser)
- ◆ Mai exista si alte optiuni la directiva FRAMESET. Pentru aceasta se poate consulta o documentatie detaliata de HTML.
- ◆ Scopul acestui capitol este doar o trecere in revista a limbajului HTML, ceea ce am si facut

Bibliografie

1. VA, Do you need to learn HTML?
<http://www.craftedweb.com/website-development/do-you-need-to-learn-html/>
2. Head First HTML with CSS & XHTML
<http://books.google.ro/books?id=589iSbn9WYkC>
3. Tutorialele lui Joe Barta
(o copie este la adresa info.cs.pub.ro)