

Curs 5

2017/2018

# Programarea aplicațiilor web

# Recapitulare

---

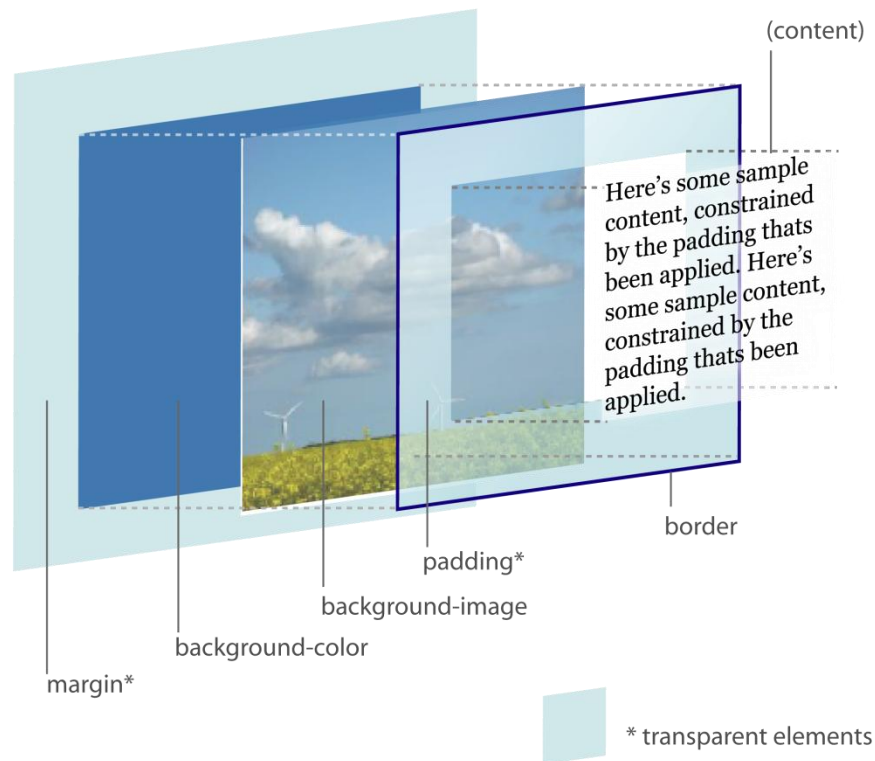
Capitolul II

**CSS**

# CSS Box Model

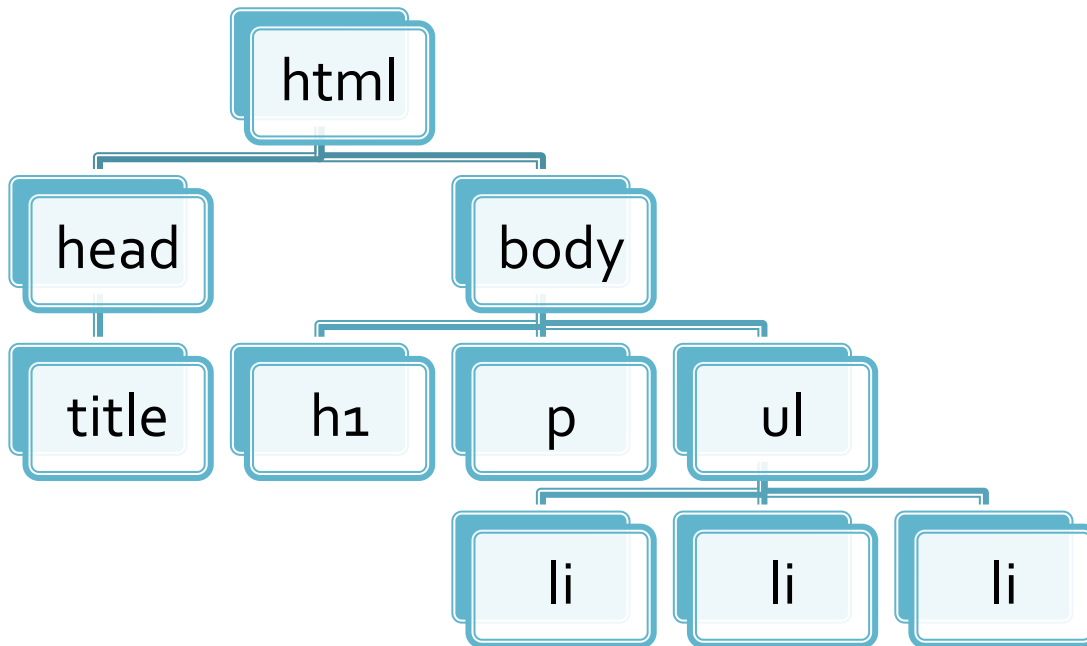
- Orientat in jurul conceptului de "cutie" – Box model

THE CSS BOX MODEL HIERARCHY



# DOM

- DOM – Document Object Model: structura de tip graf



```
<html>
  <title>pagina mea</title>
  <body>
    <h1>Compozitori:</h1>
    <p>
    <ul>
      <li> elvis costello
      <li> johannes brahms
      <li> georges brassens
    </ul>
  </body>
</html>
```

Hypertext PreProcessor

**PHP**

# Integrare

```
539         <td><?php echo $row['Documente'];?>&nbsp;  </td>
540         <td><?php echo $row['user_creat'];?>&nbsp;  </td>
541         <td class="smaller"><a href="control_lot.php?id=<?php echo $row['ID_LOT'];?>">dezactiveaza</a><br /><a href=
"control_lot.php?id=<?php echo $row['ID_LOT'];?>">modifica</a></td>
542     </tr>
543     <?php $index++;
544     } while ( $row = mysql_fetch_assoc($result));?>
545 </table><?php
546 }
547 else
548 {
549     echo "<p>Nu exista loturi active</p>";
550 }
551 ?>
552
553 <p class="title">Loturi inactive</p>
554 <?php
555 $query = "SELECT l.*, c.`nume_user` AS `user_creat`
556         FROM `lot` AS l
557         LEFT JOIN `users` AS c ON (l.`User`=c.`id_user`)
558         WHERE l.`Activ` = 0 ORDER BY l.`ID_LOT` DESC";
559 $result = mysql_query($query);
560 $total=0;
561 if ($result && (mysql_num_rows($result) > 0))
562     {
563     $total=mysql_num_rows($result);
564     $row = mysql_fetch_assoc($result);
565     }
566 if ($total>0)
567 {?>
568 <table align="center">
569     <tr class="lista_titlu">
570         <td align="center">Nr. </td>
```

# Variante de integrare

- Toate variantele ofera aceeasi sursa HTML pentru browser
- E **recomandata** cea care lasa structura HTML nemodificata si doar datele dinamice sunt rezultatul procesarii
- Codul HTML + PHP e interpretat mult mai elegant in editoarele WYSIWYG

```
<h2>Rezultate comanda</h2>  
<?php echo '<p>Comanda receptionata</p>';?>
```

```
<h2>Rezultate comanda</h2>  
<p><?php echo 'Comanda receptionata';?></p>
```

```
<?php echo '<h1>Magazin online XXX SRL</h1>';?>  
<?php echo '<h2>Rezultate comanda</h2>';?>  
<?php echo '<p>Comanda receptionata</p>';?>
```

```
<?php  
echo '<h1>Magazin online XXX SRL</h1>';  
echo '<h2>Rezultate comanda</h2>';  
echo '<p>Comanda receptionata</p>';  
?>
```



# PHP – tipuri de date

```
$var = expresie
```

- Controlul variabilelor se face automat, “on the fly”
  - Daca \$var nu era definita anterior, in urma atribuirii se defineste de tipul dat de rezultatul expresiei
  - Daca \$var era definita, de un anumit tip (oarecare), in urma atribuirii devine de tipul dat de rezultatul expresiei
  - La finalizarea executiei script-ului se elimina variabila din memorie (automat)

# PHP – tipuri de date

- tipul de date este in totalitate dependent de ceea ce se stocheaza
- PHP reactualizeaza tipul pentru a putea primi ceea ce se stocheaza

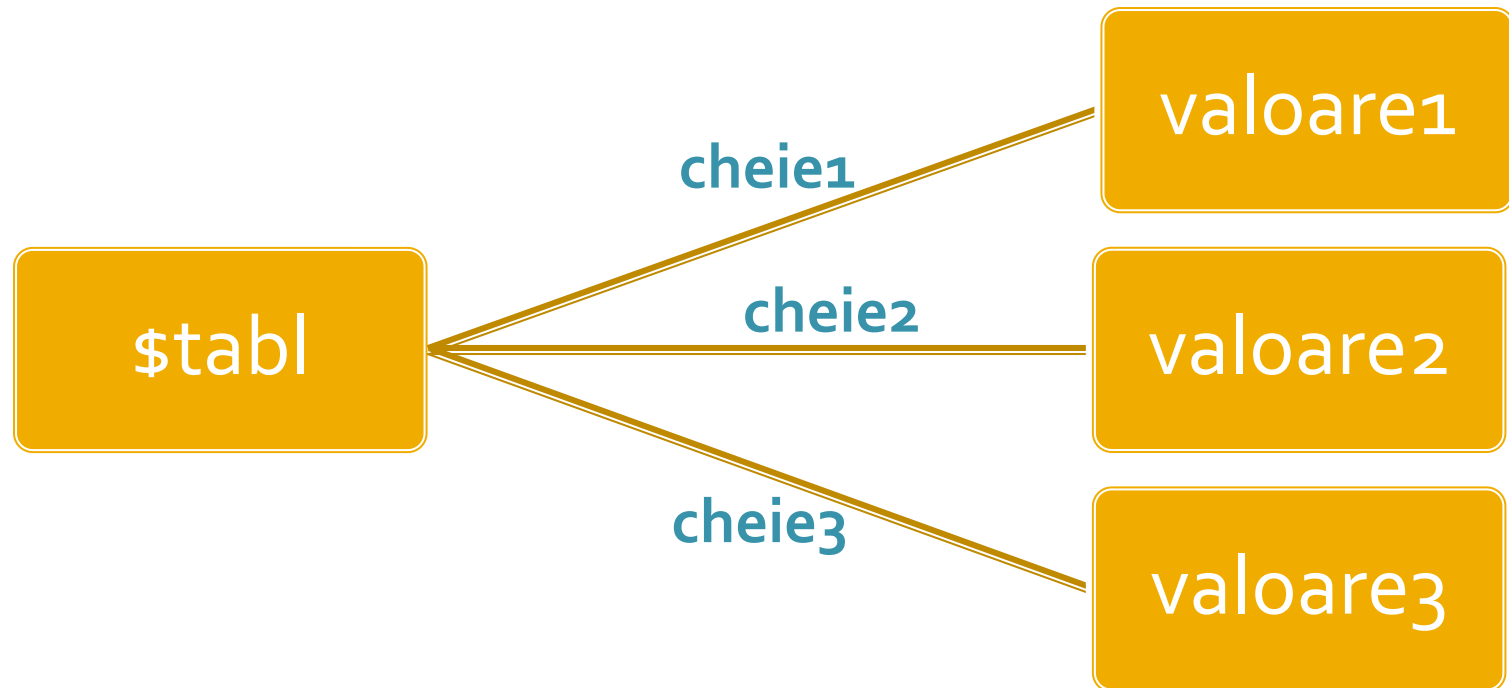
```
<?php
echo $variabila ; // tip Null, neinitializat – valoare NULL (doar)
$variabila = "0"; // $variabila tip string (ASCII 48)
$variabila += 2; // $variabila tip integer (2)
$variabila = $variabila + 1.3; // $variabila tip float (3.3)
$variabila = 5 + "10 obiecte"; // $variabila tip integer (15)
$var2=5; // $var2 tip integer (5)
$variabila=$var2."10 obiecte"; // $variabila tip string "510 obiecte"
?>
```

# Tablouri in PHP

- tabloul este tipul de variabila care asociaza **valori** unor **chei**
- spre deosebire de C, Basic, **cheile nu sunt** obligatoriu numere **intregi**, pot fi si **siruri**
- implicit cheile sunt intregi succesivi (pentru fiecare element adaugat) si primul element este 0.
- definirea unei perechi cheie / valoare
  - cheie => valoare
- definirea unui tablou
  - `$matr = array("definirea perechilor chei/valori")`
  - pereche: cheie => valoare, ...

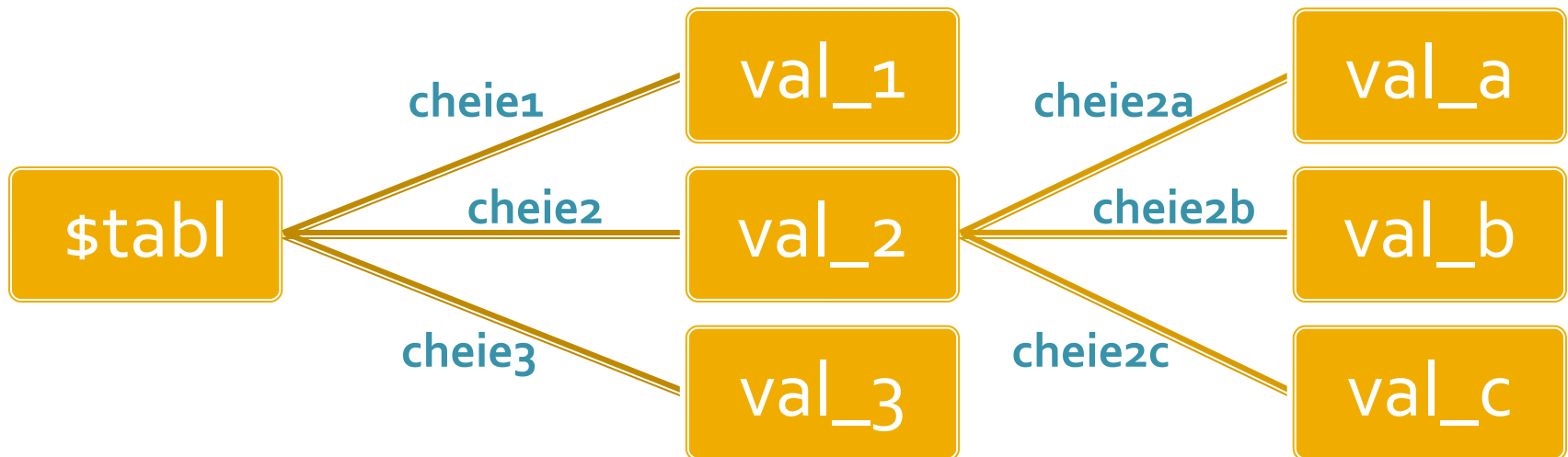
# Tablou = Arbore

- \$tabl = array(cheie1 => valoarea1, cheie2 => valoarea2, cheie3 => valoarea3)



# Tablou = Arbore

- In particular, una sau mai multe dintre din valori poate fi la randul ei tablou, ducand la **ramificarea** arborelui
- \$tabl = array(cheie1 => val\_1, **cheie2 => array(cheie2a => val\_a, cheie2b => val\_b, cheie2c => val\_c), cheie3 => val\_3)**



# Elemente de control – foreach

- `foreach (array_expression as $key => $value) statement`
- `foreach (array_expression as $value) statement`
- iterarea prin fiecare element al tabloului
- la fiecare element variabila declarata in instructiune variabila locala `$key` ofera acces la cheia curenta iar variabila locala `$value` ofera acces la valoarea asociata
- `foreach()` lucreaza cu o **copie** a tabloului deci tabloul original nu va fi modificat prin schimbarea continutului variabilelor `$key` si `$value`

# Elemente de control – foreach

```
$matr = array (  
    "fruits" => array("a" => "orange", "b" => "banana", "c" => "apple", "ultim"),  
    "numbers" => "in loc de numere",  
    "holes" => "in loc de ce era"  
);  
foreach ($matr as $scheie => $continut)  
    echo "matr[".$scheie."]=".$continut."<br />";
```

```
matr[fruits]=Array  
matr[numbers]=in loc de numere  
matr[holes]=in loc de ce era
```

# Laborator 4 – modificat!

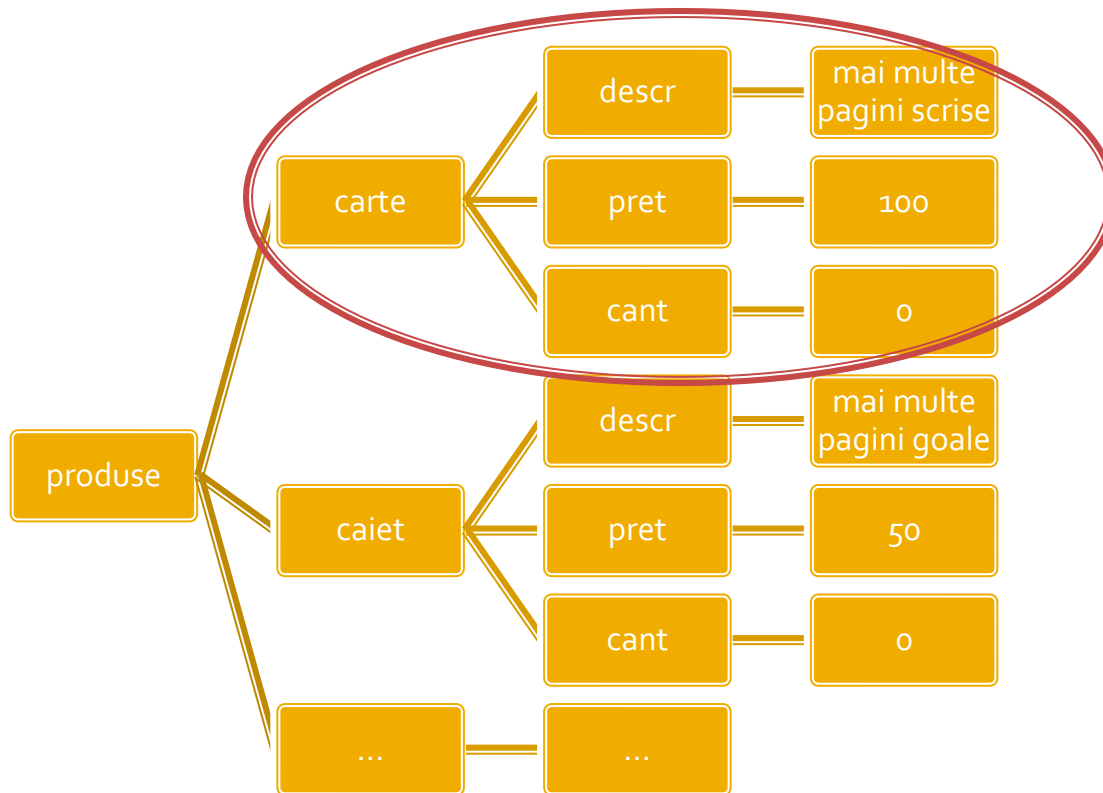
- Sa se continue magazinul virtual cu:
  - ~~■ produsele sunt grupate pe **categorii** de produse~~
  - ~~■ sa prezinte utilizatorului o lista de categorii de produse pentru a alege~~
  - sa prezinte utilizatorului o lista de produse si preturi in categoria aleasa
  - lista de produse si preturi se citeste dintr-un **fișier**
  - se preia comanda si se calculeaza suma totala
- Optional
  - ~~■ se creaza o pagina prin care vanzatorul poate **modifica** preturile si produsele~~



# Laborator 4 – Mod de lucru

- 1. Se introduce in surse facilitatea template
- 2. Se modifica sursele pentru lucru cu matrici
- 3. Se modifica sursele pentru a citi datele de pe disc
  - **anterior** se creaza fisierul text sau:
  - **o singura data** se salveaza datele (S72)
- 4. ~~Se introduce structura suplimentara, categorie~~
  - ~~se **creaza pagina** de selectie a categoriei, din care se va merge in lista de produse (utilizare \$\_GET S76)~~
- 5. ~~Optional: Se creaza o pagina care sa permita modificarea fisierului~~
  - numai pret/descriere, fara adaugare/schimbare produse

# Laborator 4 – Matrice produse



# Template

antet.php

```
<html>
<head>
<title>Magazin online Firma X
SRL</title>
</head>
<body bgcolor="#CCFFFF"><?php
define('PRET_CARTE',100);
define('PRET_CAJET',50);
define('PRET_PENAR',150);
define('PRET_STILOU',125);
define('PRET_CREION',25);
//orice cod comun PHP
?><table width="600" border="0"
align="center">
<tr><td></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFFCC">
<h1>Magazin online Firma X SRL</h1>
```

subsol.php

```
</td></tr>
</table>
</body>
</html>
```

\*.php

```
<?php require('antet.php');?>
<h2>Lista Produse</h2>
<table border="1">
...
</table>
<?php require('subsol.php');?>
```

# Utilizare template - recomandat

- sectiunile repetabile pot fi mutate intr-un fisier separat si introduse cu `require()`
- se identifica zonele comune

```
<html>
<head>
<title>Magazin online Firma X SRL</title>
</head>
<body bgcolor="#CCFFFF">
<table width="600" border="0" align="center">
<tr><td></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFFCC">
Continut
</td></tr>
</table>
</body>
</html>
```

# Utilizare template - recomandat

antet.php

```
<html>
<head>
<title>Magazin online Firma X
SRL</title>
</head>
<body bgcolor="#CCFFFF"><?php
define('PRET_CARTE',100);

//orice cod comun PHP

?><table width="600" border="0"
align="center">
<tr><td></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFFCC">
<h1>Magazin online Firma X SRL</h1>
```

subsol.php

```
</td></tr>
</table>
</body>
</html>
```

\*.php

```
<?php require('antet.php');?>
<h2>Lista Produse</h2>
<table border="1">
...
</table>
<?php require('subsol.php');?>
```

# Utilizare template

- antet.php
  - orice cod de structura (HTML) comun
  - orice cod de aplicatie comun (PHP) – aproape toate paginile dintr-o aplicatie au nevoie de:
    - acces la date
    - verificare drepturi de acces
    - definitii constante
    - definire/incarcare date **din** sesiunea de lucru (\$\_SESSION)
- subsol.php
  - orice cod de structura (HTML) comun
  - orice cod de aplicatie comun (PHP) – de obicei mai redus:
    - salvare date **in** sesiunea de lucru (\$\_SESSION)

# Format Text – Utilizatori umani

- se pot utiliza si alte caractere pentru separare
  - esential: sa nu apara in date
  - TAB are efect vizual si in fisiere text

carte	mai multe pagini scrise legate	100	0
caiet	mai multe pagini goale legate	75	0
penar	loc de depozitat instrumente de scris	150	0
stilou	instrument de scris albastru	125	0
creion	instrument de scris gri	25	0

# Exemplu

- fiecare produs e caracterizat de:
  - nume
  - pret
  - (eventual) descriere
  - cantitate comandata
- tabloul asociat:  
`$produse["produs"] = tablou["caracteristici"]`

```
$produse = array ( 'carte' => array ("descr" => "mai multe pagini scrise", "pret" => 100, "cant" => 0),  
  'caiet' => array ("descr" => "mai multe pagini goale", "pret" => 50, "cant" => 0),  
  'penar' => array ("descr" => "loc de depozitat instrumente", "pret" => 150, "cant" => 0),  
  'stilou' => array ("descr" => "instrument de scris albastru", "pret" => 125, "cant" => 0),  
  'creion' => array ("descr" => "instrument de scris gri", "pret" => 25, "cant" => 0)  
  );
```



**Continuare**

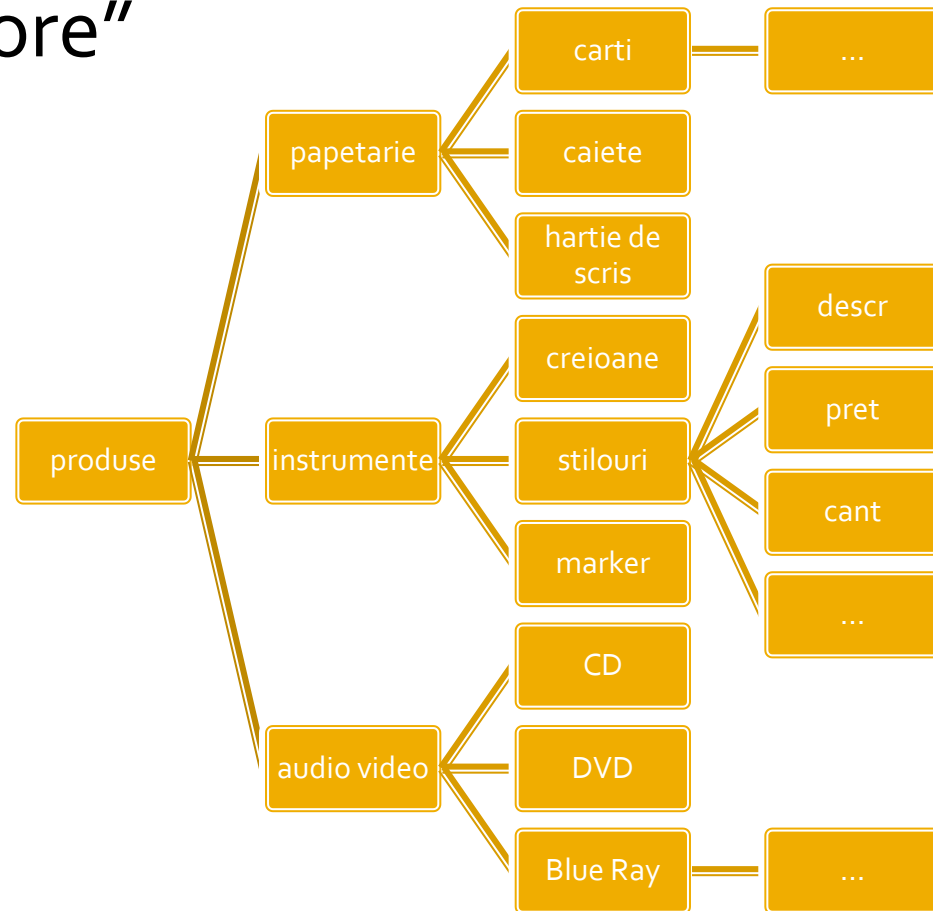
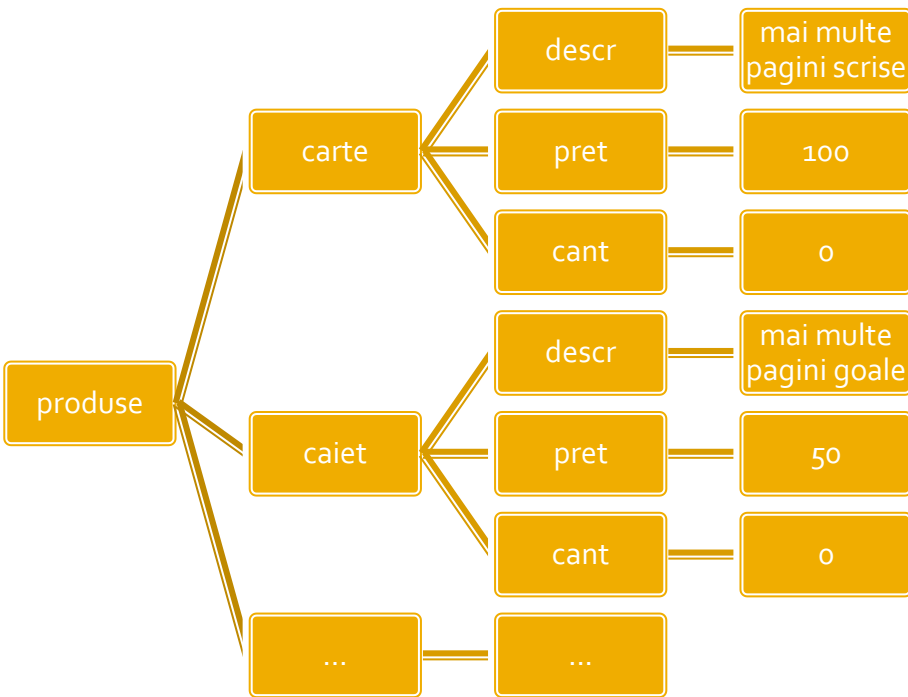
---

# Scrierea datelor pe disc

- Structura datelor este total dependentă de algoritmul utilizat
- Modificările sunt dificil de realizat și implică schimbarea fișierului de date și schimbarea codului în întregime
- Exemplu: gruparea elementelor pe categorii
  - `$produse["produs"] = ["caract."]`
  - `$produse["categorie"]["produs"] = ["caract."]`
- index-ul suplimentar introdus transformă o matrice bidimensională în una tridimensională
  - dificil de implementat cu fișiere text (caract/linii – 2D)

# Matrici

- adaugare nivel in "arbore"



# Rezolvare

- modificarea fisierului text cu introducerea categoriei ca prim parametru in fisier

carte	mai multe pagini scrise legate	100	0	
caiet	mai multe pagini goale legate	75	0	
penar	loc de depozitat instrumente de scris	150	0	
stilou	instrument de scris albastru	125	0	
papetarie	carte	mai multe pagini scrise legate	100	0
papetarie	caiet	mai multe pagini goale legate	75	0
instrumente	penar	loc de depozitat instrumente de scris	150	0
instrumente	stilou	instrument de scris albastru	125	0

# Dezavantaj

- date redundante: numele categoriei apare listat pentru fiecare produs in parte
- sensibilitate la erori si nevoie de precizie in realizarea fisierului
  - numele categoriei trebuie scris exact de fiecare data
  - o mica greseala ("instrument" in loc de "instrumente") are ca efect crearea unei categorii suplimentare
- Daca numarul de produse e mare schimbarea e dificil de realizat manual, si complicat de realizat prin cod
- Produsele care nu apartin nici unei categorii nu vor putea fi diferite

# Cod

- Codul ramane in mare parte acelasi
- Cateva modificari sunt necesare

```
$matr=file("produse.txt");  
foreach ($matr as $linie)  
    {  
        $valori=explode("\t",$linie,4);  
        $produse[$valori[0]]=array ("descr" => $valori[1], "pret" => $valori[2], "cant" => $valori[3]);  
    }
```


```
$matr=file("produse.txt");  
foreach ($matr as $linie)  
    {  
        $valori=explode("\t",$linie,5);  
        $produse[$valori[0]] [$valori[1]]=array ("descr" => $valori[2], "pret" => $valori[3], "cant" =>  
$valori[4]);  
    }
```

# Acces la date

- Utilizare, cu doua bucle foreach

```
$index=1;
foreach ($produse as $prod => $detalii) //primul indice in $produse imi da produsul
    {?>
<tr><td><?php echo $index;?></td><td><?php echo ucfirst(strtolower($prod));?></td><td><?php echo
$detalii['descr'];?></td><td align="center"><?php echo $detalii['pret'];?></td></tr>
<?php $index++;
    }
```

```
$index=1;
foreach ($produse as $categ => $lista_categ) //primul indice in $produse imi da categoria
    foreach ($lista_categ as $prod => $detalii) //al doilea indice in $produse imi da produsul
        //din categoria stabilita cu primul indice
            {?>
<tr><td><?php echo $index;?></td><td><?php echo $categ;?></td><td><?php echo
ucfirst(strtolower($prod));?></td><td><?php echo $detalii['descr'];?></td><td align="center"><?php
echo $detalii['pret'];?></td></tr>
<?php $index++;
        }
```



# Rezultat

**Magazin**

**Firma X SRL**

## Magazin online Firma X SRL

### Lista Produse

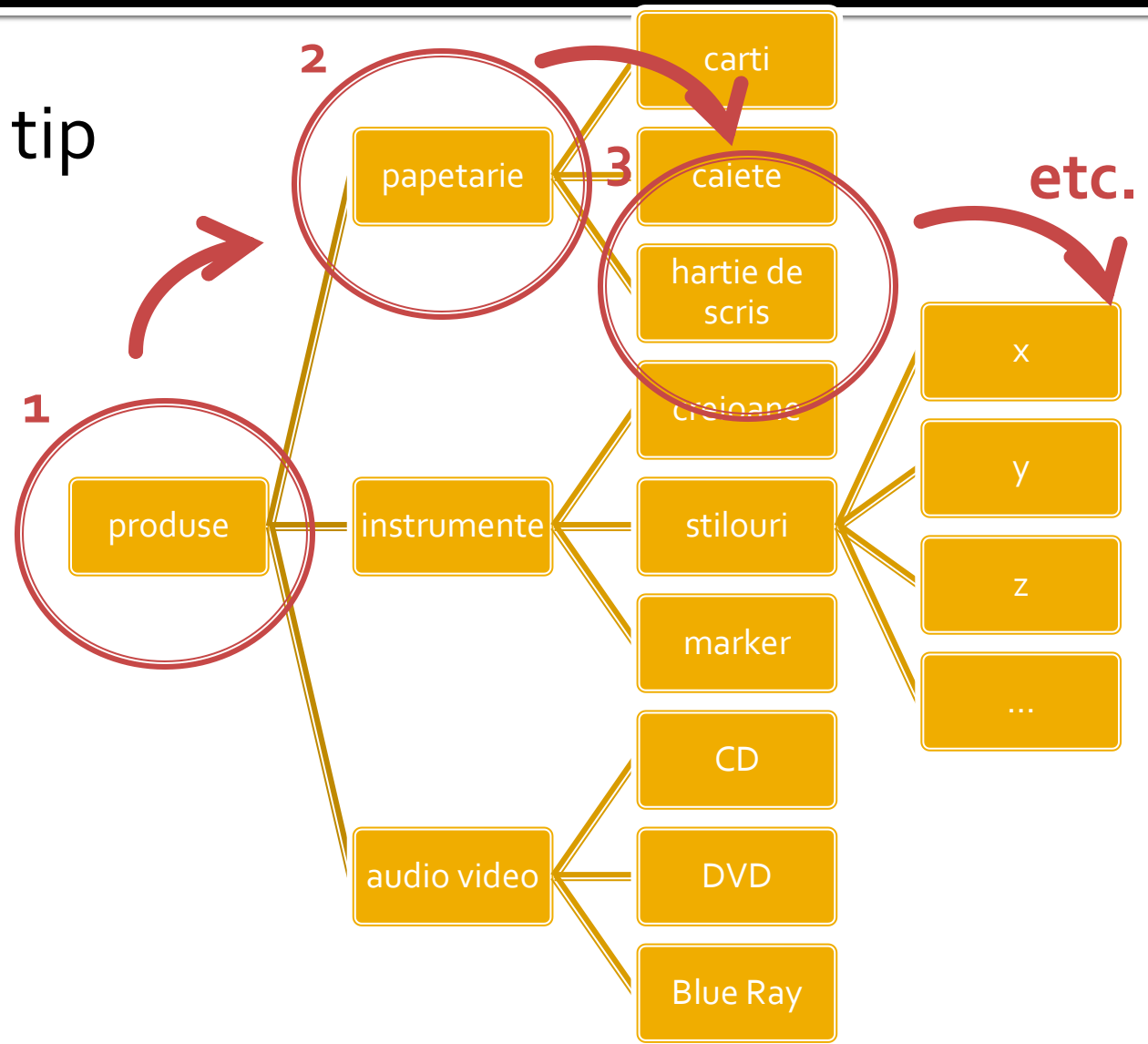
Nr.	Categorie	Produs	Descriere	Pret
1	papetarie	Carte	mai multe pagini scrise legate	100
2	papetarie	Caiet	mai multe pagini goale legate	75
3	papetarie	Hartie scris	mai multe pagini goale NElegate	50
4	instrumente	Penar	loc de depozitat instrumente de scris	150
5	instrumente	Stilou	instrument de scris albastru	125
6	instrumente	Creion	instrument de scris gri	25
7	instrumente	Marker	instrument de scris verde	25
8	audio-video	Cd	canta	50
9	audio-video	Dvd	vizual	100
10	audio-video	Blue ray	vizual extrem	500

[Comanda](#)



# Matrici

- structura tip "arbore"



# Accesul la metode externe de stocare eficiente a datelor

# XML

- XML - eXtensible Markup Language
- O forma a SGML - Standard Generalized Markup Language (ISO 8879:1986 SGML)
- O metoda de a descrie structura si importanta datelor si continutul lor fara a da indicatii despre afisare
- XSLT - XSL Transformations (Extensible Stylesheet Language) limbaj de conversie a XML in alte tipuri de documente XML cu sau fara reprezentare grafica

# HTML/XHTML vs XML

- XML
  - proiectat pentru a **descrie** datele
  - orientat spre **continutul** datelor respective
  - o metoda de a transmite informatiile **independent** de platforma si hardware
- HTML/XHTML
  - proiectat pentru a **afisa** datele
  - orientat spre **forma** pe un ecran a datelor respective
  - o metoda de a **afisa uniform** datele indiferent de platforma si hardware

# XML

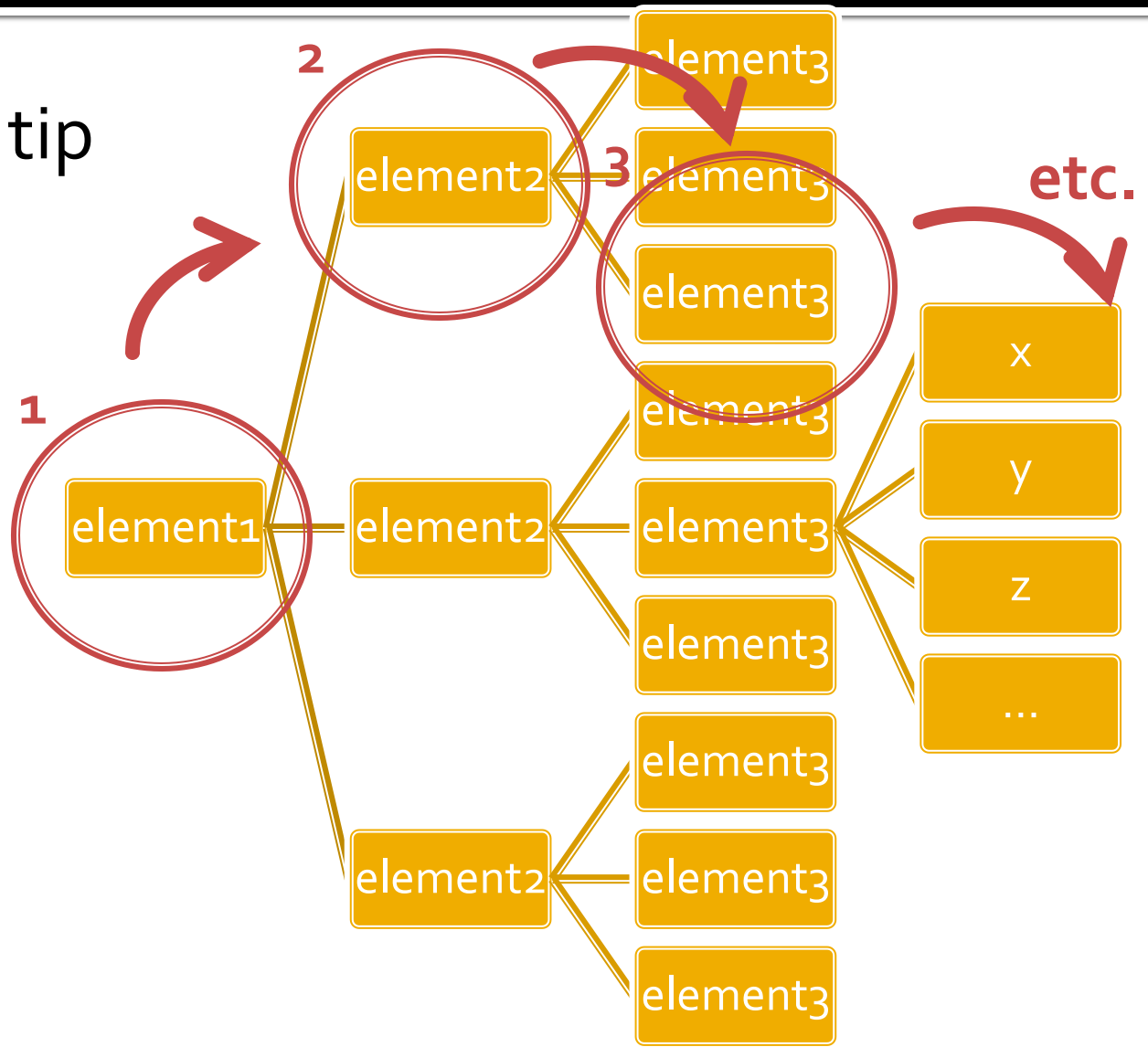
- In conceptie asemanator cu XHTML
  - etichete XHTML ("tag" - EN)
  - elemente XML ("element" - EN) descrise de etichete ("tag" - EN)
- Elementele XML accepta attribute (similar XHTML)
- Conceptele de scriere a documentului similar XHTML
- Diferenta majora:
  - HTML – etichetele si attributele sunt predefinite si orientate spre modalitatea de afisare a datelor
  - XML – etichetele de identificare a elementelor si attributele sunt la latitudinea creatorului documentului, introducand **structura** in date

# XML

- `<element1 atribut1 = "valoare" ... >`
  - `<element2 atribut2 = "valoare">`
    - `<element3>...</element3>`
    - ...
  - `</element2>`
  - ...
- `</element1>`
- ...
- `<element1>`
- `</element1>`

# XML

- structura tip "arbore"



# Exemplu XML

```
<?xml version="1.0" encoding="utf-8"?>
<gallery titlu="Photomagic" thumbDir="./fotografii/thumbnails/" imageDir="./fotografii/">
  <category nume="VIATA">
    <image>
      <desc>curiozitate</desc>
      <img>foto33.jpg</img>
      <thumb>foto33TH.jpg</thumb>
    </image>
  </category>
  <category nume="NUNTA">
    <image>
      <desc>asteptare</desc>
      <img>foto132.jpg</img>
      <thumb>foto132TH.jpg</thumb>
    </image>
  </category>
</gallery>
```



# Avantaje

- **Redundanta**
  - fiecare element XML trebuie introdus complet
  - aceasta permite detectia si corectarea facila a erorilor
- **Auto descriptiv**
  - XML este un limbaj bazat pe text, insesi elementele si attributele descriu datele
  - usor de citit/corectat pentru utilizatori umani
- **Generalitate**
  - orice fisier XML poate fi citit de orice aplicatie XML
  - anumite aplicatii necesita o anumita structura a datelor

# Reguli XML

- Aproape orice caracter UNICODE poate fi utilizat
- 107000 caractere, 90 scrieri diferite
- exceptii:
  - &lt; <
  - &gt; >
  - &amp; &
  - &quot; "
  - &apos; '

# Reguli XML

- etichetele de definire a elementelor **trebuie** închise sau elementul declarat ca vid
  - `<descriere> ... </descriere>`
  - `<descriere></descriere>`
  - `<descriere />`
- atributele **trebuie** scrise între ghilimele
  - `<categorie nume="papetarie">`
- etichetele și atributele sunt **Case Sensitive**
  - **gresit (tehnic)** -> `<descriere> ... </Descriere>`
  - **gresit (logic)** -> `<descriere> ... </descriere>`  
`<DESCRIERE> ... </DESCRIERE>`

# Reguli XML

- Documentele XML creaza o structura ierarhica foarte stricta
- Nu sunt permise etichete suprapuse
  - `<x><y></y></x>` → permis
  - `<x><y></x></y>` → interzis
- Trebuie sa existe un singur element radacina care sa le cuprinda pe toate celelalte
  - similar cu `<html></html>`

# Structura unui document XML

- prima linie – definitia tipului de document
  - `<?xml ... ?>`
  - `<?xml version="1.0" encoding="utf-8"?>`
- element radacina
  - `<radacina> ...[elemente]... </radacina>`

# XML Concepte

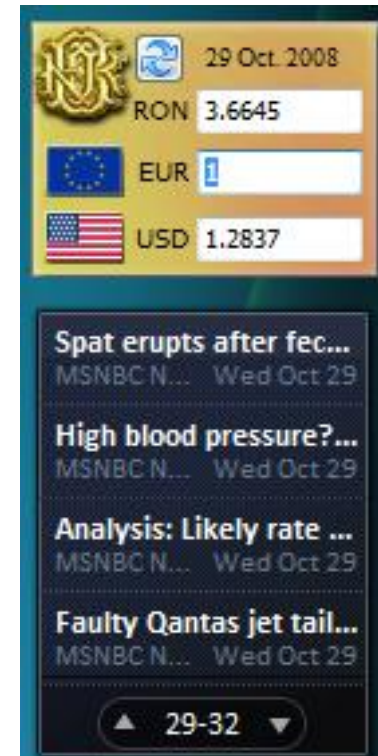
- comentariile pot fi introduse oriunde in interiorul documentului cu conditia sa fie in exteriorul oricarui element
  - similare cu comentariile HTML: intre `<-- si -->`
- Sectiuni de date neinterpretate
  - intre `<![CDATA[ si ]]>`
  - pentru a putea introduce date care ar putea contine caracterele interzise
    - cod
    - date binare oarecare

# Exemple XML

# RSS




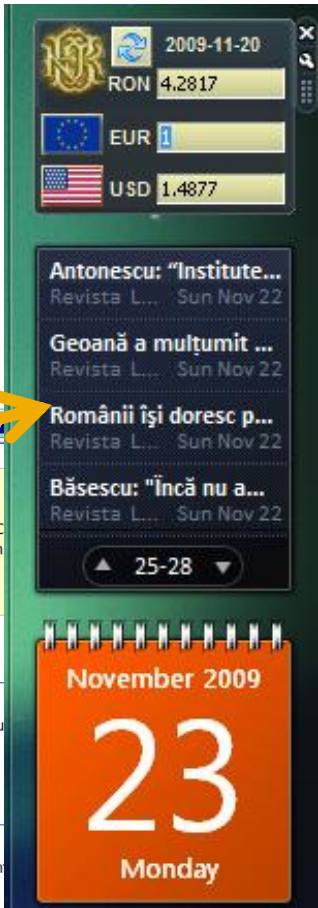
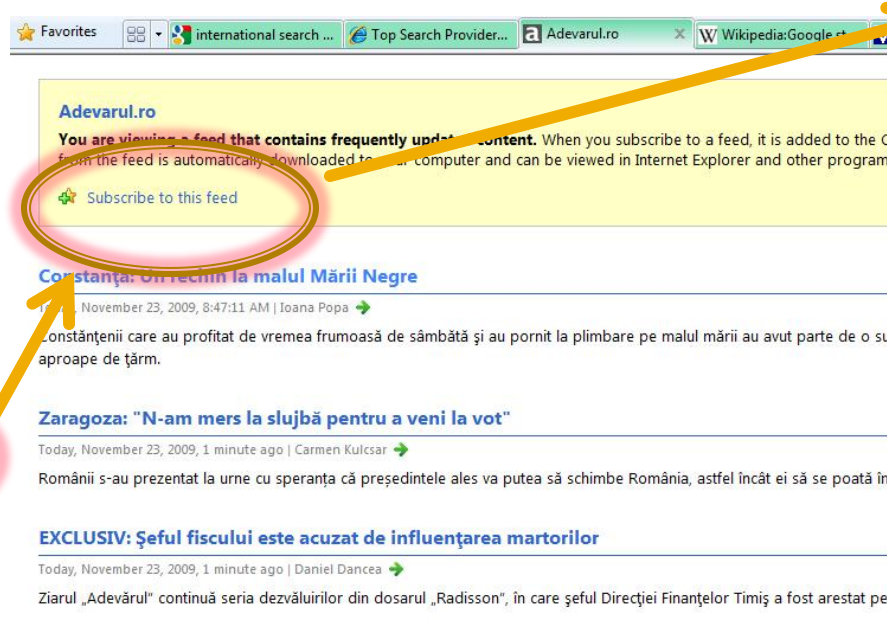
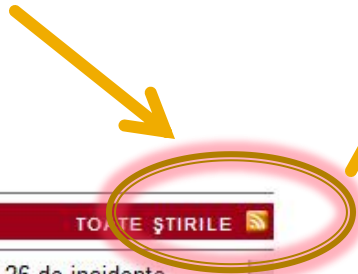
- RSS – Really Simple Syndication
- Format Web utilizat pentru a face disponibile utilizatorilor sau altor server-e informatii frecvent modificate
  - stiri
  - postari pe forum





# RSS

- Accesul la fluxurile de date
  - identificarea canalelor 
  - inscrierea la canal ("Subscribe to this feed")
  - aplicatie pentru afisarea informatiilor



08:42 | Poliția a înregistrat 26 de incidente electorale în județ-Revista presei locale

08:41 | Rezultatele etapei a 13-a din Italia

BNR Politică monetară Stabilitate financiară Supraveghere Sisteme de plăți Monede și bancnote Publicații **Statistică** Legislație

Seturi de date  
Cursul de schimb  
Serii statistice  
Info financiar  
Rate dobânzi de politică monetară  
Operațiuni de piață monetară  
Titluri de stat  
Statistică monetară și financiară  
Raport inflație  
Rapoarte asupra stabilității financiare

Statistică > Seturi de date > Cursul de schimb

### Cursul de schimb Leu (RON)

	13 mar. 2017	14 mar. 2017	15 mar. 2017	16 mar. 2017
Dolarul austriacian	AUD 3,2525	3,2525	3,2434	
Leva bulgărească	BGN 2,3283	2,3272	2,3228	
Dolarul canadian	CAD 3,1725	3,1801	3,1779	

BANCA NAȚIONALĂ A ROMÂNIEI  
Obiectivul fundamental al BNR este asigurarea și menținerea stabilității prețurilor.

### Link-uri utile

- Serii zilnice - ultimele 10 înregistrări
- Serii lunare, trimestriale și anuale
- Baza de date interactivă (permite vizualizarea și exportul seriilor istorice - zilnice, lunare, anuale)
- Pentru programatori: Preluarea cursului valutar prin formate XML

BNR Politică monetară Stabilitate financiară Supraveghere Sisteme de plăți Monede și bancnote Publicații **Statistică** Legislație

Seturi de date  
Cursul de schimb  
Serii statistice  
Info financiar  
Rate dobânzi de politică monetară  
Operațiuni de piață monetară  
Titluri de stat  
Statistică monetară și financiară  
Raport inflație  
Rapoarte asupra stabilității financiare  
Contururile naționale financiare  
Indicatori plăți  
Balanța de plăți - BPM6  
Finanțe publice  
Situatiile financiare ale BNR

Statistică > Seturi de date > Cursul de schimb > Cursurile în format XML

### Cursurile pieței valutare în format XML

Pentru diseminarea cursurilor valutare, BNR pune la dispoziția programatorilor fișiere de date în format XML. Actualizarea acestora se face în timp real, în fiecare zi bancară, imediat după ora 13:00.

```
<?xml version="1.0" encoding="UTF-8"?>
- <DataSet xsi:schemaLocation="http://www.bnr.ro/xsd nbrfxrates.xsd" >
  - <Header>
    <Publisher>National Bank of Romania</Publisher>
    <PublishingDate>2017-03-17</PublishingDate>
    <MessageType>DR</MessageType>
  </Header>
  - <Body>
    <Subject>Reference rates</Subject>
    <OrigCurrency>RON</OrigCurrency>
    - <Cube date="2017-03-17">
      <Rate currency="AED">1.1542</Rate>
      <Rate currency="AUD">3.2609</Rate>
      <Rate currency="BGN">2.3342</Rate>
      <Rate currency="BRL">1.3600</Rate>
      <Rate currency="CAD">3.1807</Rate>
      <Rate currency="CHF">4.2598</Rate>
      <Rate currency="CNY">0.6140</Rate>
      <Rate currency="CZK">0.1690</Rate>
      <Rate currency="DKK">0.6141</Rate>
      <Rate currency="EGP">0.2335</Rate>
```

### FIȘIERE DE DATE ÎN FORMAT XML

- Serii de înregistrări zilnice:
  - cea mai recentă
  - ultimele 10 înregistrări
- Serii de înregistrări anuale:
  - 2017
  - 2016
  - 2015
  - 2014
  - 2013
  - 2012
  - 2011
  - 2010
  - 2009
  - 2008
  - 2007
  - 2006
  - 2005

Conținutul sau structura acestora se poate modifica fără preaviz.

Traficul generat de aceste fișiere poate genera un volum de trafic excesiv și poate cauza probleme de trafic redundant. IP-urile care generează un volum de trafic excesiv vor fi blocate temporar pentru a remedia situația pe adresa de e-mail a webmaster.

Doar utilizatorii autorizați să efectueze operațiuni pe piața valutară.

Operațiunile efective de schimb valutar sau înregistrări contabile.

# XML in PHP

# Acces prin cod PHP la date XML

- **Biblioteci** PHP de acces la XML
  - DOM XML
  - XML Parser (Reader, Writer)
  - Simple XML

# Simple XML

- `$xml = simplexml_load_file('cale');`
- `$xml = simplexml_load_string($string);`
- Functiile care creaza un obiect din clasa predefinita **SimpleXMLElement** prin incarcarea si interpretarea documentului sau sirului XML
- obiectul creat (`$xml`) contine elementul radacina
  - exemplul anterior → `$xml` va contine `<gallery></gallery>`

# Simple XML

- fiecare **tip** de element descendent creaza o proprietate a clasei cu acel nume
  - proprietatea clasei (descendent) e o matrice de obiecte **SimpleXMLElement** daca nu e unicul descendent
  - matricea e caracterizata de elementele implicite de indexare (index intreg, pornind de la 0) dar e recomandata iteratia folosind `foreach()`

# Exemplul anterior

- elementul <gallery> contine un numar mare de descendentii de un singur tip, <category>
- obiectul radacina a fost redenumit prin operatia de deschidere a fisierului
  - `$xml = simplexml_load_file('cale');`
- obiectul `$xml` va contine ca proprietate
  - `$xml->category`
  - `$xml->category` este o matrice deoarece nu este o singura categorie
  - descendentii se pot accesa cu
    - `$xml->category[0], $xml->category[1], ...`
    - `foreach ($xml->category as $categ_curent) { // $categ_curent obiect de tip SimpleXMLElement, aceleasi reguli }`

# Simple XML

- attributele XML pot fi accesate in obiectul corespunzator ca si cum ar fi elementele unei matrici cu indice text, indicele fiind acelasi cu numele atributului cautat
- In exemplul anterior fiecare element `<category>` are un atribut `nume`
  - `<category nume="VIATA">`
  - ```
foreach ($xml->category as $categ_curent)
{
    echo $categ_curent['nume'];
}
```



# Recursivitate

- Fiecare proprietate a clasei **SimpleXMLElement** este de asemenea un obiect din clasa **SimpleXMLElement** daca mai are descendenti
  - `$xml->category[o]` este de tip obiect:  
`$xml->category[o] ->image ...`
- Proprietatea clasei contine datele elementului daca nu mai sunt alti descendenti
  - `echo $xml->category[o]->image[o]->img // foto33.jpg`
- Pentru ca proprietatea sa fie tratata ca fiind de tip string (implicit la afisare) trebuie fortata conversia
  - `if ((string)$xml->category[o]->image[o]->img=="foto33.jpg") ...`

# Modificare de cod XML

- incepand cu PHP 5.1.3.
- SimpleXMLElement::addChild(string nume, string valoare)
- SimpleXMLElement::addAttribute(string nume, string valoare)
- SimpleXMLElement::asXML('filename') scrie in fisierul 'filename' rezultatul sau il returneaza ca un sir
- schimbarea continutului unui element
  - `$xml->category[0]->image[0]->img = "foto33.jpg"`

# Laborator 5

# Rezultat

## Categorii Produse

Alegeti categoria:

| Nr. | Categorie                    | Total Produse |
|-----|------------------------------|---------------|
| 1   | <a href="#">Papetarie</a>    | 3             |
| 2   | <a href="#">Instrumente</a>  | 3             |
| 3   | <a href="#">Audio-video</a>  | 3             |
| 4   | <a href="#">Calculatoare</a> | 3             |
| 5   | <a href="#">Jucarii</a>      | 2             |

Total produse: 14

## Magazin online Firma X SRL

### Realizati comanda

| Nr. | Produs   | Pret | Cantitate                      |
|-----|----------|------|--------------------------------|
| 1   | Carti    | 100  | <input type="text" value="1"/> |
| 2   | Caiete   | 50   | <input type="text" value="2"/> |
| 3   | Penare   | 150  | <input type="text" value="1"/> |
| 4   | Stilouri | 125  | <input type="text" value="0"/> |
| 5   | Creioane | 25   | <input type="text" value="0"/> |

Trimite

## Magazin online Firma X SRL

### Rezultate comanda

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

# Plan aplicatie

- Pe masura ce aplicatia paraseste un fir liniar de executie este necesara introducerea unui plan (graf) al aplicatiei
- Cumparator
  - citirea datelor de pe disc se realizeaza in **antet.php**, comun pentru toate fisierele

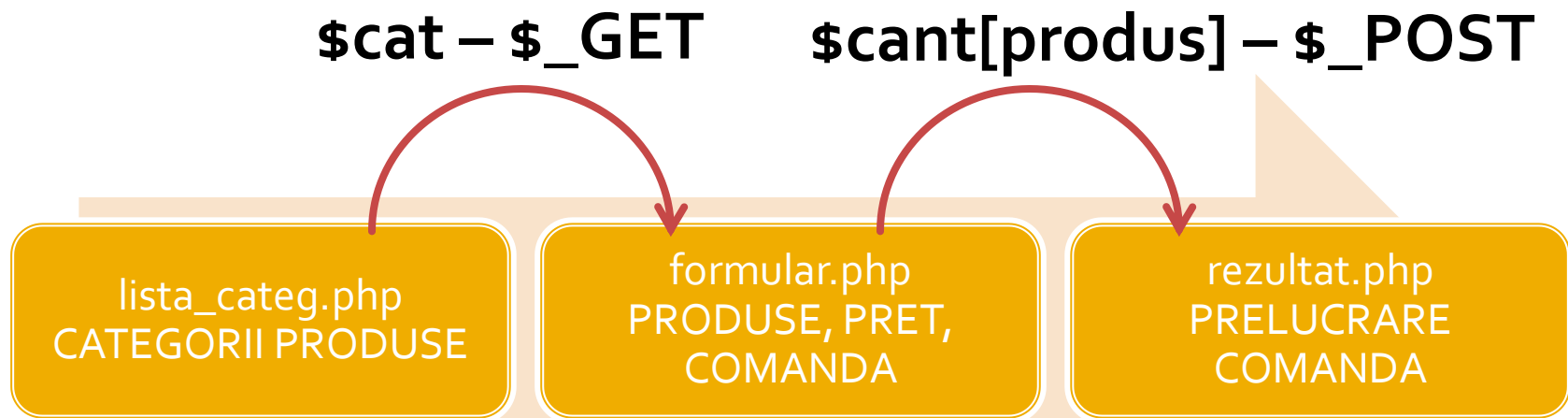
lista\_categ.php  
CATEGORII PRODUSE

formular.php  
PRODUSE, PRET,  
COMANDA

rezultat.php  
PRELUCRARE  
COMANDA

# Plan aplicatie

- Planul aplicatiei trebuie sa cuprinda si informatii relative la:
  - **ce date** se transmit intre diferitele pagini
  - **cum** se transmit datele intre pagini

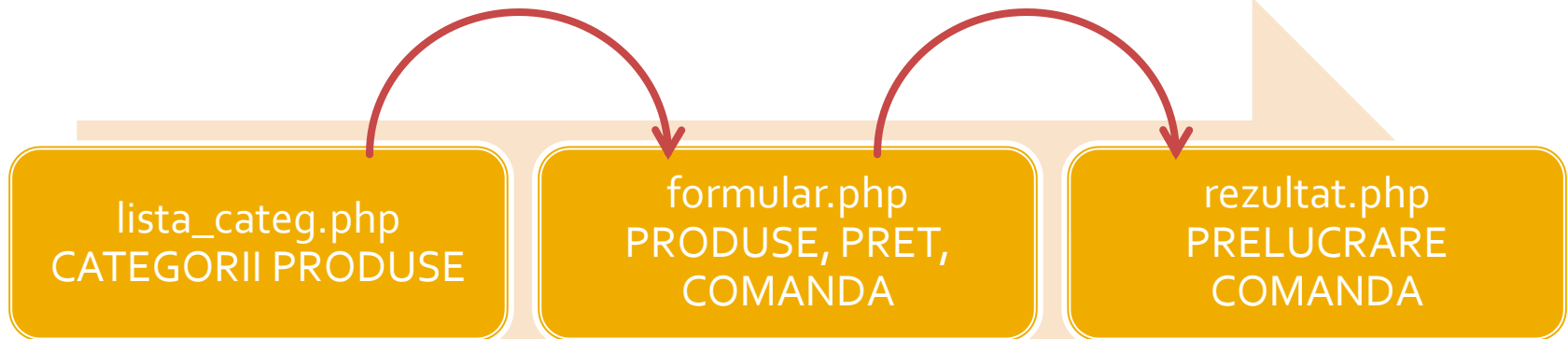


# Plan aplicatie

- Planul aplicatiei – Exemplu
  - lista de categorii va contine “**link-uri active**” deci transmiterea unei singure variabile se face cu **\$\_GET**
  - formularul de comanda transmite date multiple incluse intr-o forma deci transmiterea se face cu **\$\_POST**
- Alegerea \$\_GET/\$\_POST are implicatii:
  - atat in pagina care transmite datele
  - cat si in pagina care le receptioneaza

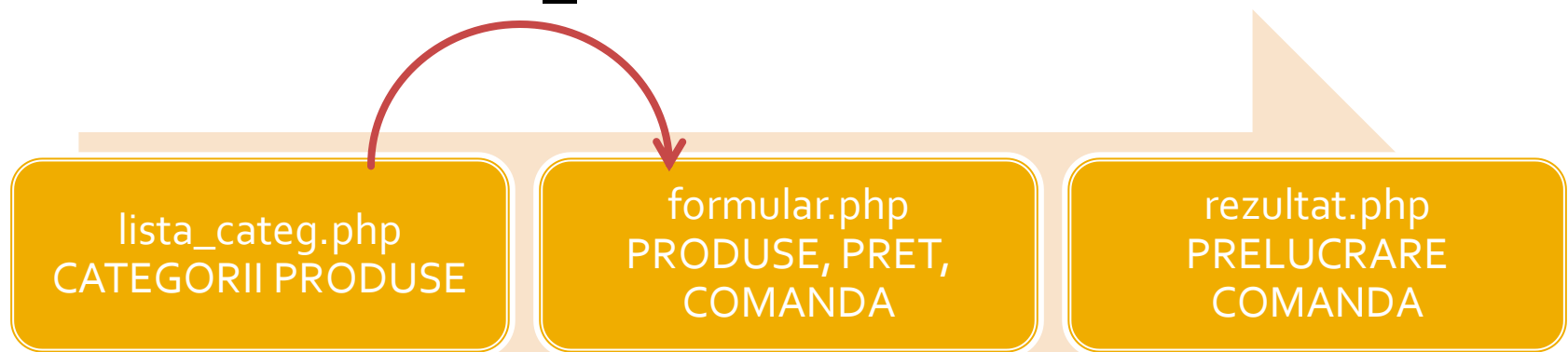
**\$cat – \$\_GET**

**\$cant[produs] – \$\_POST**



# Link-uri active

- folosite pentru a transmite o **informatie**
- in `lista_categ.php`
  - `<a href="lista_prod.php?categ=<?php echo $cat;?>"> <?php echo $cat;?> </a>`
- are efect in `formular.php`
  - `$_GET['categ']="valoarea $cat corespunzatoare"`  
**\$cat – \$\_GET**





# Laborator 4 – modificat!

- Sa se continue magazinul virtual cu:
  - ~~■ produsele sunt grupate pe **categorii** de produse~~
  - ~~■ sa prezinte utilizatorului o lista de categorii de produse pentru a alege~~
  - sa prezinte utilizatorului o lista de produse si preturi in categoria aleasa
  - lista de produse si preturi se citeste dintr-un **fisier**
  - se preia comanda si se calculeaza suma totala
- Optional
  - ~~■ se creaza o pagina prin care vanzatorul poate **modifica** preturile si produsele~~

# Laborator 5

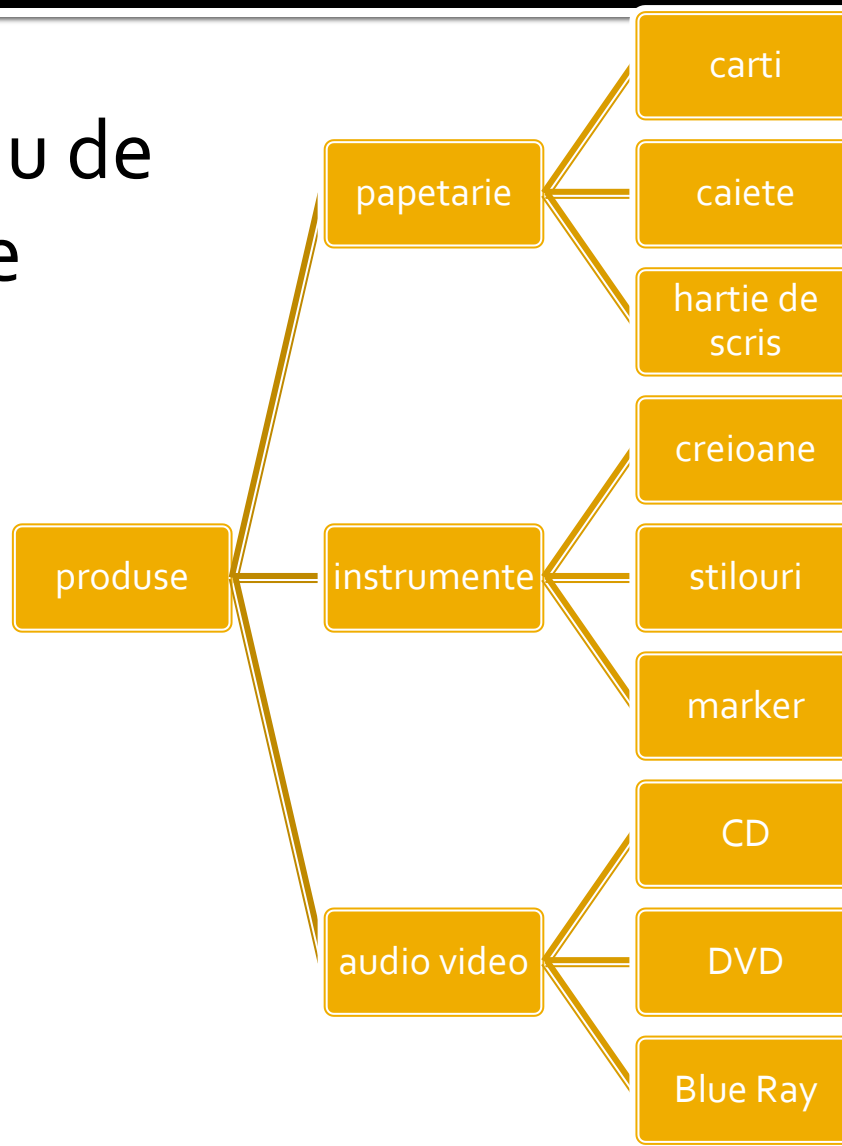
- Sa se continue magazinul virtual cu:
  - produsele sunt grupate pe **categorii** de produse
  - sa prezinte utilizatorului o lista de categorii de produse pentru a alege
  - sa prezinte utilizatorului o lista de produse si preturi in categoria aleasa
  - lista de produse si preturi se citeste dintr-un **fișier**
  - se preia comanda si se calculeaza suma totala
- Optional
  - se creaza o pagina prin care vanzatorul poate **modifica** preturile si produsele

# Laborator 5 – Mod de lucru

- 1. Se introduce in surse facilitatea template
- 2. Se modifica sursele pentru lucru cu matrici
- 3. Se modifica sursele pentru a citi datele de pe disc (C5 – fisier text)
  - **anterior** se creaza fisierul text sau:
  - **o singura data** se salveaza datele (C5 – S72)
- 4. Se introduce structura suplimentara, categorie
  - se **creaza pagina** de selectie a categoriei, din care se va merge in lista de produse (utilizare \$\_GET – S103)
- 5. Lista de produse si preturi se citeste dintr-un fisier **XML**
- 6. Optional: Se creaza o pagina care sa permita modificarea fisierului text/XML
  - numai pret/descriere, fara adaugare/schimbare produse

# Laborator 5

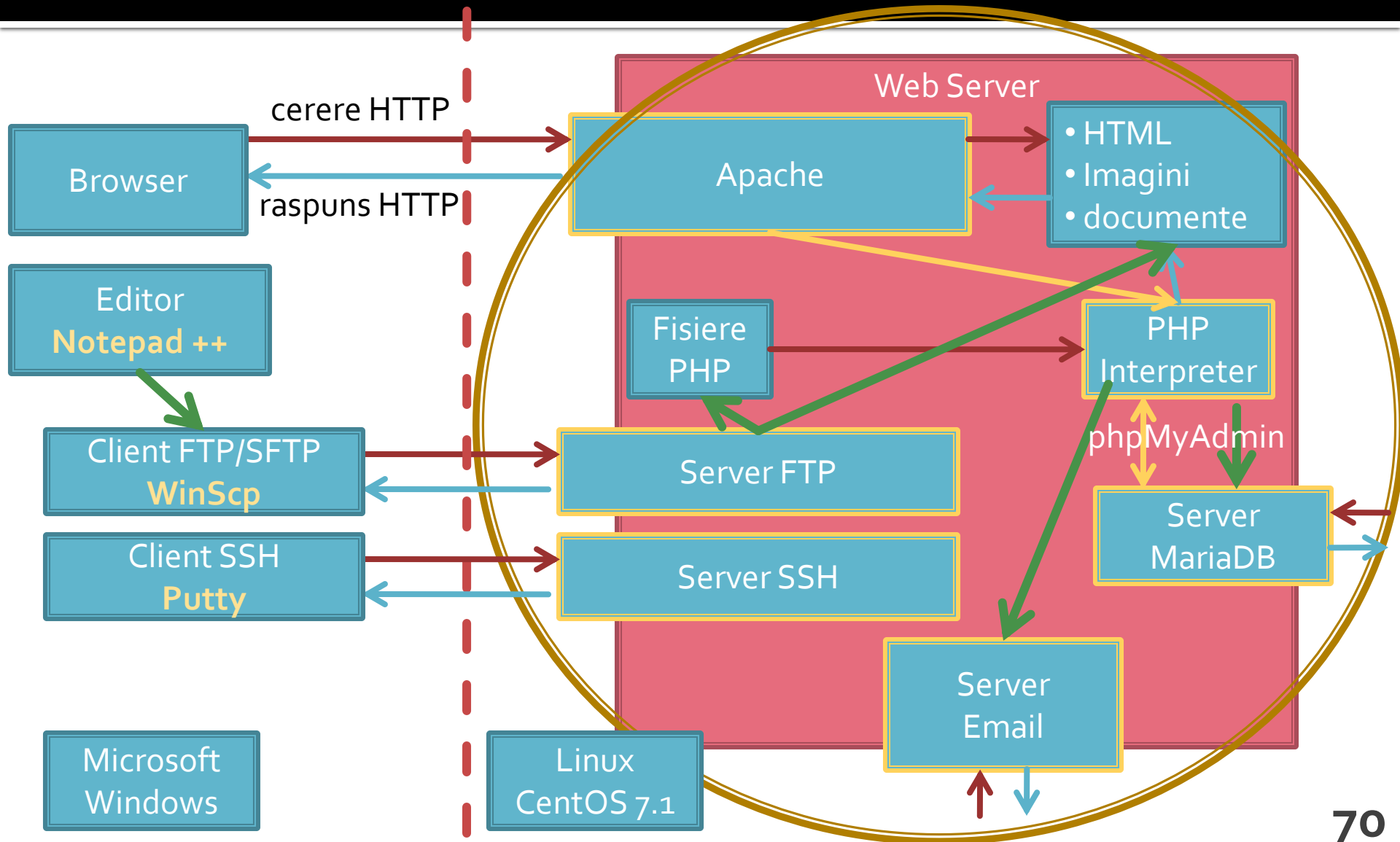
- exemplu de grupare



# Laborator 5 – Mod de lucru

- Se continua lucrul la aplicatie (L4)
- Se recomanda laboratorul **asincron** – S10
- Se poate folosi fisierul cu surse cpypaste.txt (site-<http://rf-opto.etti.tuiasi.ro>)

# Utilizare LAMP



# Utilizare LAMP

**Categorii Produse**

Alegeti categoria:

| Nr. | Categorie                    | Total Produse |
|-----|------------------------------|---------------|
| 1   | <a href="#">Papetarie</a>    | 3             |
| 2   | <a href="#">Instrumente</a>  | 3             |
| 3   | <a href="#">Audio-video</a>  | 3             |
| 4   | <a href="#">Calculatoare</a> | 3             |
| 5   | <a href="#">Jucarii</a>      | 2             |

Total produse: 14

**Magazin online Firma X SRL**

Realizati comanda

| Nr. | Produs   | Pret | Cantitate                      |
|-----|----------|------|--------------------------------|
| 1   | Carti    | 100  | <input type="text" value="1"/> |
| 2   | Caiete   | 50   | <input type="text" value="2"/> |
| 3   | Penare   | 150  | <input type="text" value="1"/> |
| 4   | Stilouri | 125  | <input type="text" value="0"/> |
| 5   | Creioane | 25   | <input type="text" value="0"/> |

**Magazin online Firma X SRL**

Rezultate comanda

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

<input name="x" ..

\$\_POST['x']

\$\_GET['x']

Web Server

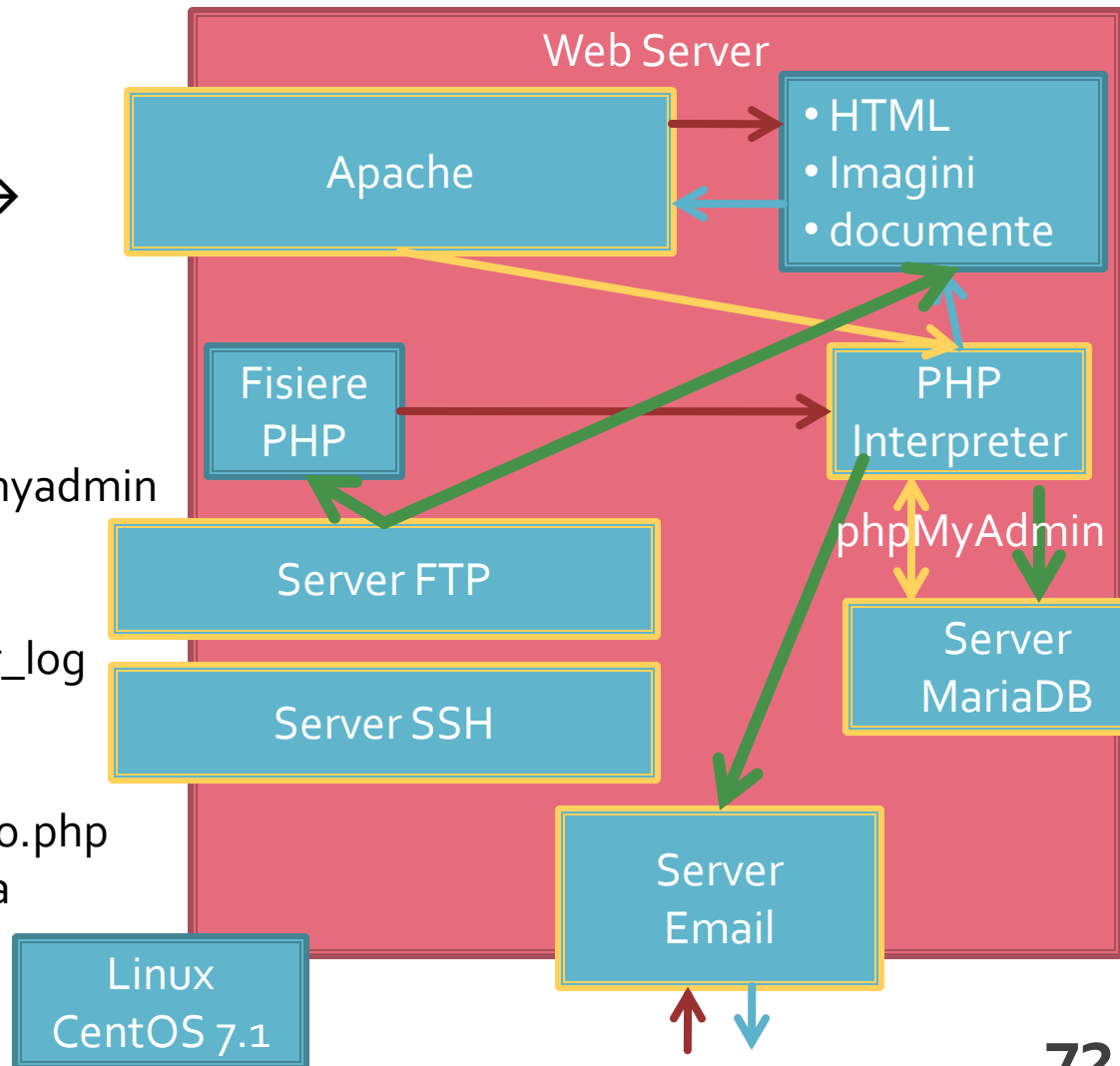
Apache

Server FTP

Server SSH

# Utilizare LAMP

1. login → root:masterrc
2. ifconfig → 192.168.30.5
3. putty.exe → 192.168.30.5 → SSH → root:masterrc (remote login)
4. [alte comenzi linux dorite]
5. FTP → Winscp → SFTP → student:masterrc@192.168.30.5
6. MySql → http://192.168.30.5/phpmyadmin → root:masterrc
7. Apache Error Log →
  - 7a. putty → nano /var/log/httpd/error\_log
  - 7b. http://192.168.30.5/logfile.php (nonstandard)
8. PHP info → http://192.168.30.5/info.php
9. daca serviciul DHCP duce la oprirea Apache: `service httpd restart`





# Faza de verificare/depanare

- Se recomanda utilizarea posibilitatii vizualizarii matricilor
  - In fisierul care receptioneaza datele
  - temporar pina la definitivarea codului
- utilizarea de cod "verbose" (manual) in etapele initiale de scriere a surselor PHP poate fi extinsa si la alte tipuri de date
  - singura (aproape) metoda de depanare(debug) in PHP
  - `<p>temp <?php echo "a=";echo $a; ?> </p>`

```
echo "<pre>";  
print_r($_POST);  
echo "</pre>";
```

# Depanare

```
echo "<pre>";  
print_r($_POST);  
echo "</pre>";
```

```
<p>temp <?php echo  
"a=";echo $a; ?> </p>
```

# Contact

- Laboratorul de microunde si optoelectronica
- <http://rf-opto.etti.tuiasi.ro>
- [rdamian@etti.tuiasi.ro](mailto:rdamian@etti.tuiasi.ro)