

Curs 4

2018/2019

Programarea aplicațiilor web

Introducere

PAW

- Programarea aplicațiilor web
 - An V RC
 - 2C/1L/1P

Nota

- An V
 - 33% E
 - 66% Aplicatii
 - 33% L (0%)
 - 33% P (66%)

Documentatie

- <http://rf-opto.etti.tuiasi.ro> + WWW



The screenshot displays the website for RF-OPTO at ETTI. The header includes the ETTI logo, the text 'RF-OPTO', and the University of Technical Sciences 'Gheorghe Asachi' Iasi logo. A navigation menu contains 'Main', 'Courses', 'Master', 'Staff', 'Research', and 'Students'. Below the menu, there are links for 'Radiocommunication Systems', 'Microwave IC', and 'Web Design'. The main content area features the course title 'Modern Web Applications Design' and the following details:

Course: TMPAW (2014-2015)
Course Coordinator: Assist.P. Dr. Radu-Florin Damian
Code: DS507RC
Discipline Type: DOMS; Alternative, Synthesis
Credits: 7
Enrollment Year: 5, Sem. 10

Activities
Course: Instructor: Assist.P. Dr. Radu-Florin Damian, 2 Hours/Week, Specialization Section, Timetable:
Laboratory: Instructor: Assist.P. Dr. Radu-Florin Damian, 1 Hours/Week, Half Group, Timetable:

Evaluation
Type: **Examen**
A: 33.33%, (Test/Colloquium)
B: 33.33%, (Seminary/Laboratory/Project Activity)
D: 33.33%, (Homework/Specialty papers)

Grades
Not yet

Attendance
Not yet

Materials
Course Slides

CURS

I.	HTML si XHTML (recapitulare)	1 oră
II	CSS	2 ore
III	Baze de date, punct de vedere practic	1 oră
IV	Limbajul de interogare SQL	4 ore
V	PHP - HyperText Preprocessor	8 ore
VI	XML - Extended Mark-up Language si aplicatii	4 ore
VII	Conlucrare intre PHP/MySql, PHP/XML, Javascript/HTML	2 ore
VIII	Exemple de aplicatii	6 ore
	Total	28 ore

rf-opto.etti.tuiasi.ro

- http://rf-opto.etti.tuiasi.ro/master_it.php
- Laborator 2011-2012
 - Exercitii/Performanta MySql
 - optional

Laborator 2

HTML

- se creaza o pagina care sa arate ca in schita alaturata
- forma paginilor:
 - tabel
- Se experimenteaza diversele elemente de interactiune cu utilizatorul

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	

Suplimentar

- forma din schita alaturata
- forma paginilor:
 - tabel controlat prin CSS

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	
	Copyright (cu alta culoare fundal)	

Analiza critica

- design?
 - in aplicatiile web forma **este importanta**
 - nu trebuie sa fie inovativa ci familiara
 - “Don’t make me think!”

Design 1

```
<html>
<head>
<title>Magazin online Firma X SRL </title>
</head>
<body>
<table width="100%" border="0"
bgcolor="#CCFFFF">
<tr><td rowspan="2" width="*"></td>
<td width="600"></td>
<td rowspan="2" width="*"></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFACC">
Continut
</td></tr>
</table>
</body>
</html>
```

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	



tabel

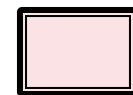
Design 1



Design 2

```
<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="#FFFACC">
Continut
</td></tr>
</table>
</body>
</html>
```

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	



tabel

Design 2



Design 3

index.html

```
<html>
<head>
<title>Magazin online Firma X SRL</title>
<link rel="stylesheet" href="stil.css"
type="text/css" />
</head>
<body>
<div class="antet"></div>
<div class="continut">
Continut
</div>
</body>
</html>
```

stil.css

```
body { background-color: #CCFFFF;}
.antet { background-image:url(images/antet.gif);
background-repeat:no-repeat;
height: 100px;
width: 600px;
margin-top: 0px;
margin-right: auto;
margin-left: auto;}
.continut{ background-color:#FFFFCC;
height: 600px;
width: 600px;
margin-top: 5px;
margin-right: auto;
margin-left: auto;}
```


Design 3



Continuare

- capacitatea de extindere?
 - va aparea un meniu?
 - unde?
- design 1
 - tabel cu 3 coloane, numai cea centrala e folosita
 - avantaj: se creaza doua zone care ar putea primi date **daca** e nevoie
 - dezavantaj: forma (culori, dimensiuni) intercalata in continut

Continuare

- design 3
 - avantaj: forma (culori, dimensiuni) separata de continut
 - avantaj: adaugarea altor cutii (div) care sa primeasca date **daca** e nevoie si controlul formei lor se poate realiza
 - dezavantaj: alaturarea a 2 box/div mai complexa, generatoare de probleme la incepatori
 - Bibliografie: "am mai facut ceva asemanator" – dezavantajul dispare

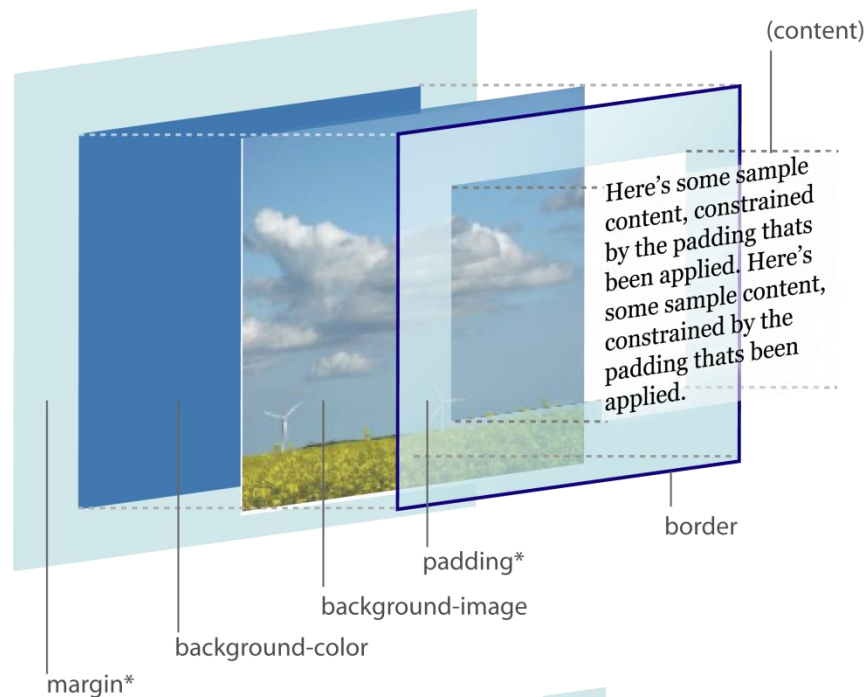
Capitolul II

CSS

CSS Box Model

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

THE CSS BOX MODEL HIERARCHY



* transparent elements

CSS Zen Garden

- <http://www.csszengarden.com/>
- un fisier html comun
- schimbarea formei permisa numai prin intermediul CSS

CSS Zen Garden – HTML original

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the

Download the sample [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless effort

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with your eyes.
Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and educate. When you visit, one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the style.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated is by placing it in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks

CSS Zen Garden



Css Zen Garden

the beauty of css design.

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

[DOWNLOAD THE SAMPLE HTML FILE AND CSS FILE](#)

The Road To Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C,

SELECT A DESIGN:

[Under the Sea!](#)
by Eric Stoltz

[Make 'em Proud](#)
by Michael McAghon and
Scotty Reifsnnyder

Consideratii generale

Aplicatii

Aplicatii

The screenshot displays the BCR online banking interface. At the top, there is a navigation bar with links for Home, Contact, English, Contact center, and a search bar. Below this is a menu with options like CLICK 24Banking, 24 Banking, Persoane fizice, Private banking, Timeri, PFA, Micro, Corporatii, IMM, Municipalitati, Despre noi, Cariere, and Presa. A secondary menu includes Lista de conturi, Conturi curente, Economisire, and Finantare.

The main content area is titled "Ordin de Plata - Creare" (Payment Order - Create). It shows a table with columns for Stare cont (Account Status), Tip (Type), Numar cont (Account Number), Sold disponibil (Available Balance), and Valuta (Currency). The table contains one entry: "Activ" (Active), "Conturi curente" (Current Accounts), a redacted account number, a redacted available balance, and "RON" (Romanian Leu).

Below the table, there is a "Detalii" (Details) section with dropdown menus for "Sablon personal" (Personal Template) and "Sablon furnizor" (Provider Template), and a text input field for "IBAN beneficiar" (Beneficiary IBAN). A "Verificare IBAN" (Verify IBAN) button is also present.

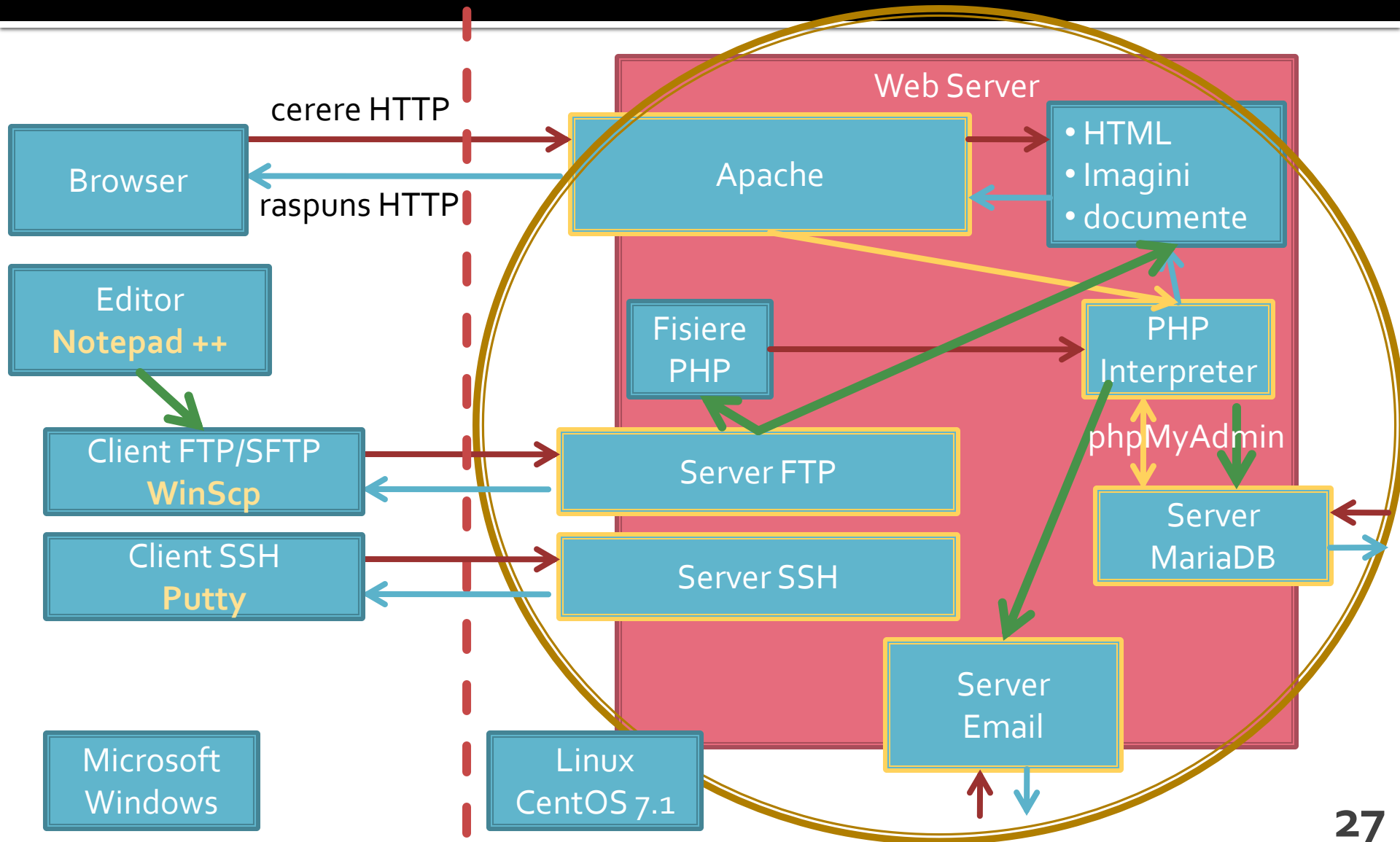
A "Nota:" (Note) section states: "Pentru a ordona o plata, va rugam fie sa introduceti IBAN beneficiar, fie sa selectati un sablon personal sau pentru furnizori. Atentie : Transferurile se pot efectua doar catre conturi deschise in aceeași valuta!" (To place an order, please either enter the beneficiary IBAN, or select a personal template or provider template. Attention: Transfers can only be made to accounts opened in the same currency!).

On the left side, there is a "Favorite" menu with options like "Ordin de Plata", "Istoricul tranzactiilor", "Lista ordine de plata", "Sabloane", "Plata repetitiva", "Schimb Valutar", "Cumpara produse", "Deschidere cont curent", "Constituire depozite la termen", "Deschidere cont de economii", "Cumparare CDD", "Ataseaza card de debit", "Aplica pentru un credit", "Curs valutar", "Mesaje(4)", "Contul meu de CLICK 24Banking (Favorite)", and "Demo Click 24 Banking".

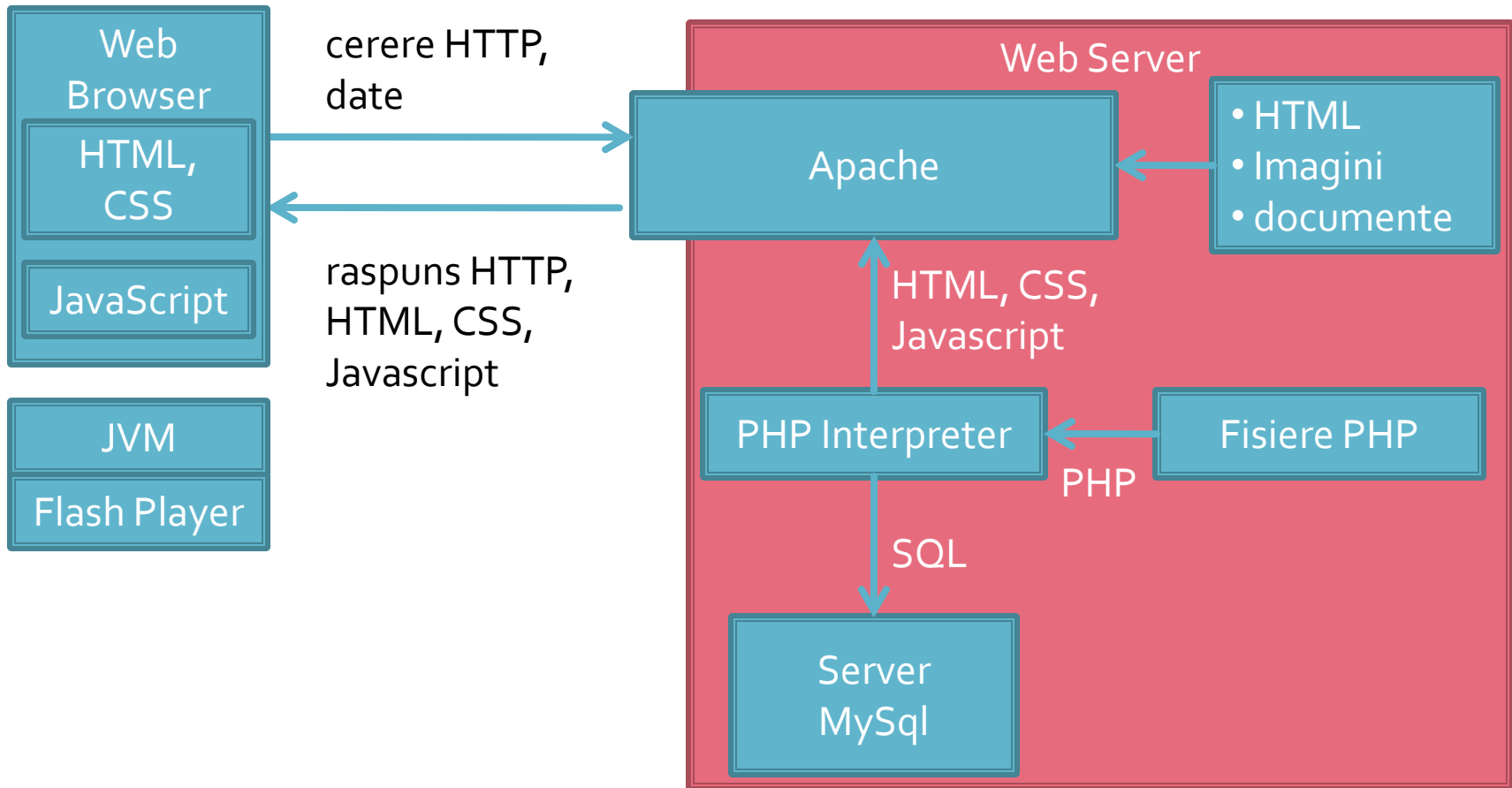
At the bottom, there are several promotional tiles: "Tarife si comisioane" (Rates and Fees), "Intrebari si Raspunsuri" (Questions and Answers), "24 Banking", and "Contact center".

The footer contains the text: "BANCA COMERCIALA ROMANA - SOCIETATE ADMINISTRATA IN SISTEM DUALIST, Bucuresti, B-dul Regina Elisabeta nr.5, Sector 3 | mentiuni legale - © 2008 BCR SA - Toate drepturile rezervate | site map".

LAMP



Client/Server Scripting



client
side
scripting

server
side
scripting

Aplicatie Web

- presupune prelucrarea unor date si oferirea unui document personalizat (rezultat al datelor respective)
- datele pot fi obtinute:
 - de la utilizator
 - o sursa externa (baze de date)
 - **combinatie** utilizator/baze de date

Forme in HTML

- necesare pentru ca utilizatorul sa poate trimite date server-ului
- `<form>...</form>`
- Attribute specifice:
 - action: adresa documentului care preia datele
 - `<form action="<?php echo $_SERVER['PHP_SELF'];?>">`
 - `<form action="actiuni/fisier2.html">`
 - method: modalitatea de transmitere a datelor: post sau get
 - `<form method="post" action= ... >`

Metode de transmitere

- **post** datele sunt transmise in bloc
- **get** datele sunt atasate adresei documentului de procesare : results.php?prob=81&an=2009
- **get** trebuie folosit numai cand datele sunt "idempotente",
 - nu cauzeaza efecte colaterale
 - nu modifica starea server-ului (baze date, etc)
- se poate simula realizarea unei forme (**get**) prin scrierea corespunzatoare a link-urilor

Elemente de interactiune cu utilizatorul

- in interiorul etichetei `<form>...</form>`
 - `input`
 - `select/option`
 - `textarea`
 - `button`
- **toate** elementele vor avea un nume
 - atribut: `name=""`
 - numele va fi intalnit in uri-ul generat prin get, sau in numele variabilei trimisa prin post
- trimiterea datelor se face prin intermediul unui buton cu tipul `type="submit"`

INPUT

- `<input .../>`
- Attribute:
 - type: text | password | checkbox | radio | submit | reset | file | hidden | image | button
 - name: numele variabilei
 - value: valoarea trimisa server-ului la selectie (valoarea initiala in anumite cazuri)
 - checked/src/size/maxlength in functie de tip

Example

```
<input name="textfield" type="text" value="ceva" />
```

```
<input name="Ok" type="submit" value="Trinite" />
```

```
<label><input name="check" type="checkbox" value="5" checked />check1</label>
```

```
<label><input name="RG1" type="radio" value="a" checked="checked" />but1</label>
```

```
<label><input name="RG1" type="radio" value="b" />but2</label>
```

```
<input name="hid" type="hidden" value="6" />
```

ceva

Trinite



check1



but1



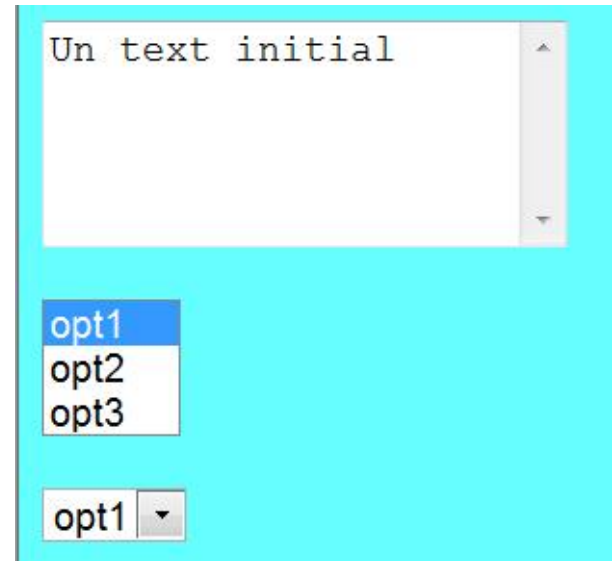
but2

TEXTAREA/SELECT

```
<textarea name="textarea" cols="20" rows="5">Un text  
initial</textarea><br /><br />
```

```
<select name="select." size="3">  
  <option value="1" selected="selected">opt1</option>  
  <option value="2">opt2</option>  
  <option value="3">opt3</option>  
</select><br /><br />
```

```
<select name="select...">  
  <option value="1" selected="selected">opt1</option>  
  <option value="2">opt2</option>  
  <option value="3">opt3</option>  
</select>
```



The image displays three visual representations of HTML form elements on a light blue background. At the top is a text area containing the text "Un text initial". Below it is a list box with three options: "opt1", "opt2", and "opt3", where "opt1" is selected and highlighted in blue. At the bottom is a dropdown menu with "opt1" selected and a downward arrow on the right.

Exemplu – forma

- Mic magazin online
- Formular de comanda cu procesarea comenzii

```
<html>
<head>
<title>Magazin online XXX SRL</title>
</head>
<body>
<h1>Magazin online XXX SRL</h1>
<h2>Realizati comanda</h2>
<form action="rezultat.html" method="post">
<table border="0">
<tr bgcolor="#cccccc"><td width="150">Produs</td><td width="15">Cantitate</td></tr>
<tr><td>Carti</td><td align="center"><input type="text" name="carti_cant" size="3" maxlength="3" /></td></tr>
<tr><td>Caiete</td><td align="center"><input type="text" name="caiete_cant" size="3" maxlength="3" /></td></tr>
<tr><td>Penare</td><td align="center"><input type="text" name="penare_cant" size="3" maxlength="3" /></td></tr>
<tr><td colspan="2" align="center"><input type="submit" value="Trimite" /></td></tr>
</table>
</form>
</body>
</html>
```

Magazin online XXX SRL

Realizati comanda

Produs	Cantitate
Carti	<input type="text" value="1"/>
Caiete	<input type="text" value="2"/>
Penare	<input type="text" value="3"/>

Exemplu – raspuns static

- fisier html
- fisierele HTML sunt doar “servite” de server
- in aparenta a existat o procesare, real **nu**

```
<html>
<head>
<title>Magazin online XXX SRL</title>
</head>
<body>
<h1>Magazin online XXX SRL</h1>
<h2>Rezultate comanda</h2>
<p>Comanda receptionata</p>
</body>
</html>
```

Magazin online XXX SRL

Rezultate comanda

Comanda receptionata

CURS

I.	HTML si XHTML (recapitulare)	1 oră
II	CSS	2 ore
III	Baze de date, punct de vedere practic	1 oră
IV	Limbajul de interogare SQL	4 ore
V	PHP - HyperText Preprocessor	8 ore
VI	XML - Extended Mark-up Language si aplicatii	4 ore
VII	Conlucrare intre PHP/MySql, PHP/XML, Javascript/HTML	2 ore
VIII	Exemple de aplicatii	6 ore
	Total	28 ore

Hypertext PreProcessor

PHP

PHP - Concepte

- limbaj interpretat – compilat “on the fly” de interpretorul PHP de pe server
- script-urile PHP contin sursele
 - exista posibilitatea pre-compilarii surselor pentru spor de viteza
 - Hip-Hop for PHP / Facebook
 - HipHop Virtual Machine / Facebook
- orientat spre aplicatii web dinamice (biblioteci)
- poate fi integrat in HTML – utilizarea tipica

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>
```

Separare cod PHP

- un fisier sursa PHP este un fisier HTML (in general) cu sectiuni de cod PHP
- interpretorul PHP cauta sectiunile pe care trebuie sa le interpreteze si interiorul lor proceseaza instructiuni ca fiind PHP
- ce se gaseste in **exteriorul** acestor sectiuni este trimis spre server-ul web **nemodificat**

Separare cod PHP

- `<?php ... ?>`
 - stil XML – implicit, disponibil intotdeauna, recomandat
- `<? ... ?>`
 - scurt, este de obicei dezactivat
 - lipsa in PHP 7
- `<script language="php"> ... </script>`
 - stil script, disponibil
- `<% ... %>`
 - stil ASP, de obicei dezactivat
 - lipsa in PHP 7

Variante de integrare

- echo ... afiseaza un text la "iesire" (echivalent cu printf() din C)
- poate realiza procesarea datelor
- in exemplu se trimite spre iesire un sir static (echivalent cu puts() din C)
- "iesire" in marea majoritate a cazurilor reprezinta datele trimise clientului de server-ul web
- "iesire" poate fi considerata de obicei:
 - documentul curent
 - pozitia curenta

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>';  
?>
```

Continuare

PHP – instructiuni

- instructiunile PHP trebuie sa se termine cu ;
 - exceptie: se poate omite ; la sfarsitul blocului de cod php: ... `echo 'ceva' ?>`
- instructiunile pot sa fie scrise pe acelasi rand (fara trecerea la linia noua)
 - `echo 'ceva1'; echo 'ceva2'; ...`
 - nu este in general recomandat

PHP – comentarii

- comentariile in PHP respecta regulile C si Perl
- la sfarsit de linie:
 - `// echo ceva; //comentariu`
 - `# echo ceva; #comentariu`
- comentariu bloc
 - `/* ... */`
 - `/* un comentariu
pe mai multe
linii */`

PHP – constante

- Ca orice limbaj de programare PHP se bazeaza pe utilizarea
 - constante
 - variabile
 - functii
- Definirea constantelor:
 - `define('PRETCARTE', 100);`
 - "case sensitive"
 - prin conventie, numai cu litere mari
 - `echo PRETCARTE; // 100`

PHP – variabile

- variabila – semnul \$ urmat de un nume
- numele e “case sensitive”
- o greseala frecventa e uitarea semnului \$
 - PHP Notice: Use of undefined constant an – assumed \$an (sau ‘an’) in D:\\Server\\
- Tipuri de date
 - scalar
 - compus
 - special

PHP – tipuri de date

- scalar
 - boolean
 - integer
 - float (double)
 - **string**
- compus
 - array
 - object
- special
 - resource
 - NULL

PHP – tipuri de date

- **declararea** variabilelor **nu** e necesara decat cand se declara un domeniu de definitie (variabile globale)
 - `global $a, $b;`
`$c=$a+$b;`
- eliberarea memoriei nu este necesara, se face automat la terminarea executiei

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"
?>
```

PHP – operatori

- In general similari celor din C/C++
- Operatori
 - Aritmetici
 - Atribuire
 - Bit
 - Comparare
 - Incrementare/Decrementare
 - Logici
 - **Sir**

PHP – operatori

- Aritmetici
 - $-$a$ – Negare
 - $$a + b – Adunare
 - $$a - b – Scadere
 - $$a * b – Inmultire
 - $$a / b Impartire
 - $$a \% b Modulo (rest)
- Sir
 - **$$a.b – Concatenare sir a si sir b**

PHP – operatori

■ Atribuire

- `$a=$b`
- `$a+=$b` ($a=a+b$)
- `$a-=$b` ($a=a-b$)
- `$a/=$b` ($a=a/b$)
- `$a*=$b` ($a=a*b$)
- `$a%=$b` ($a=a\%b$)
- `$a.= $b` ($a=a$ concatenat b - siruri)

PHP – operatori

- Operatori la nivel de bit
 - similari celor din C
 - `~, &, |, ^, <<, >>`
- Operatori logici
 - ofera rezultat boolean true/false
 - similari celor din C
 - `&&, ||, !`
 - suplimentar
 - `and, or, xor` – echivalenti dar de prioritate mai mica
 - `$a=55/0 or die('impartire prin 0');`

PHP – operatori

- Operatori de comparare
 - ofera rezultat boolean true/false
 - similari celor din C
 - == , != , > , < , <> , >= , <=
 - suplimentar
 - === identic, valoare egala **SI** de acelasi tip
 - !== “neidentic”, valoare diferita **SAU** de tipuri diferite

Precedenta operatorilor

non-associative	clone new	clone and new
left	[array()
non-associative	++ --	increment/decrement
right	~ - (int) (float) (string) (array) (object) (bool) @	types
non-associative	instanceof	types
right	!	logical
left	* / %	arithmetic
left	+ - .	arithmetic and string
left	<< >>	bitwise
non-associative	< <= > >= <>	comparison
non-associative	== != === !==	comparison
left	&	bitwise and references
left	^	bitwise
left		bitwise
left	&&	logical
left		logical
left	? :	ternary
right	= += -= *= /= .= %= &= = ^= <<= >>=	assignment
left	and	logical
left	xor	logical
left	or	logical
left	,	many uses

Precedenta operatorilor PHP7

non-associative	clone new	clone and new
left	[array()
non-associative	++ --	increment/decrement
right	~ - (int) (float) (string) (array) (object) (bool) @	types
non-associative	instanceof	types
right	!	logical
left	* / %	arithmetic
left	+ - .	arithmetic and string
left	<< >>	bitwise
non-associative	< <= > >= <>	comparison
non-associative	== != === !== <=>	comparison
left	&	bitwise and references
left	^	bitwise
left		bitwise
left	&&	logical
left		logical
right	??	comparison
left	? :	ternary
right	= += -= *= /= .= %= &= = ^= <<= >>=	assignment
left	and	logical
left	xor	logical
left	or	logical
left	,	many uses

Exemplu de separare cod php

Raspuns dinamic

- `<form action="rezultat.php" method="post">`

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

Magazin online XXX SRL

Rezultate comanda

Comanda receptionata

```
<
<
<
</head>
<body>
<h1>Magazin online XXX SRL</h1>
<h2>Rezultate comanda</h2>
<p>Comanda receptionata</p>
</body>
</html>
```

PHP – Functii

- conceptual similare celor din C/C++
- functiile nu trebuie declarate inainte de a fi folosite
- numele functiilor este “case-insensitive”
- un mare numar de functii cu utilitate directa in aplicatiile web exista in bibliotecile PHP
- unele biblioteci trebuie activate in momentul configurarii PHP
 - `extension=php_gd2.dll` (linia 639) // pentru functii de procesare grafica de exemplu
 - `extension=php_mysql.dll` (linia 651) // pentru functii de acces la baze de date MySql
 - `extension=php_mysqli.dll` (linia 652) // pentru functii de acces la baze de date MySql (**obligatoriu** de la PHP 5.6)

Utilizarea functiilor PHP

- `<form action="rezultat.php" method="post">`

`<p>Comanda receptionata la data:`

`<?php echo date('d/m/Y')." ora ".date('H:i');?></p>`

Magazin online XXX SRL

Rezultate comanda

Comanda receptionata la data: 10/03/2010 ora 13:36

`<body>`

`<h1>Magazin online XXX SRL</h1>`

`<h2>Rezultate comanda</h2>`

`<p>Comanda receptionata la data:`

`10/03/2010 ora 13:36</p>`

`</body>`

Elemente de control

- majoritatea notiunilor si sintaxei sunt similare celor din C/C++
- instructiune compusa: separata de acolade {...}
- if / else / elseif – executie conditionata

```
<?php
if ($a > $b) {
    echo "a mai mare ca b";
} elseif ($a == $b) {
    echo "a egal cu b";
} else {
    echo "a mai mic ca b";
}
?>
```

Elemente de control

- while
- do-while
- for
- switch
- return
- break
- goto

- Similare cu echivalentele C/C++

```
$i = 1;
while ($i <= 10) {
    echo $i++;
}
```

```
$i = 10;
do {
    echo $i--;
} while ($i > 0);
```

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
}
```

```
switch ($i) {
    case 0:
        echo "i este 0";
        break;
    case 1:
        echo "i este 1";
        break;
    default:
        echo "i nici 1 nici 0";
        break;
}
```

Elemente de control

- `include()`
- `require()`
- `include_once()`
- `require_once()`

- pentru inserarea **SI** evaluarea fisierului folosit ca parametru
- folosite pentru a nu multiplica sectiunile de cod comune
- `require` opreste executia script-ului curent daca fisierul parametru nu este gasit
- `..._once()` verifica daca respectivul fisier a mai fost introdus si **nu** il mai introduce inca o data

Variabile tip string

PHP – tipuri de date

- scalar
 - boolean
 - integer
 - float (double)
 - **string**
- compus
 - array
 - object
- special
 - resource
 - NULL

Variabile tip string

- Scopul final al PHP e popularea cu date (sub forma de text) a campurilor existente intr-un schelet HTML
- Ca urmare datele de tip sir de caractere (string) sunt tratate mai complex decat echivalentul C/C++
 - mai multe modalitati de definire
 - mai multe modalitati de interpretare
 - **mult** mai multe functii

Variabile tip string

- definire variabila de tip string
 - utilizare apostrof ` `
 - utilizare ghilimele " "
 - definiri tip bloc
 - heredoc <<< "X"
 - nowdoc <<<'X' (PHP>5.3.0)

Variabile tip string ` `

- apostroful ` ` e utilizat pentru definirea sirurilor primare de caractere
 - se defineste o suita de caractere
 - prelucrarile in interiorul sirului sunt reduse
 - \' reprezinta caracterul apostrof
 - \\ si \ reprezinta caracterul backslash
 - doar atat!!!

Variabile tip string ""

- ghilimelele "" sunt utilizate pentru definirea sirurilor de caractere complexe
 - prelucrarile in interiorul sirului sunt mai complexe decat echivalentul C/C++
 - caracterele ASCII speciale, identic cu C++: \n, \r, \t, \\", \v, \e, \f, \x, \u
 - \" caracterul ghilimele
 - \\$ caracterul \$
 - se interpreteaza **variabile** in interiorul sirului !!!

Variabile tip string “ ”

- caracterul \$ indica faptul ca urmeaza un nume de variabila
 - interpretorul foloseste toate caracterele care pot genera nume de variabile valide (\$x, \$x->y, \$x[y])
 - daca e nevoie de exprimare mai complexa a variabilelor (de exemplu matrici cu 2 indici x[y][z] sau cu indici neintregi) se foloseste sintaxa complexa: **{ }**

Variabile tip string " "

- sintaxa **simpla** pentru interpretarea variabilelor in interiorul sirurilor

```
<?php
$juice = "apple";

echo "He drank some $juice juice.";
// He drank some apple juice.
echo "He drank some juice made of $juices.";
// He drank some juice made of . //s character valid pentru variabile

?>
```

Variabile tip string ""

- sintaxa **simpla** pentru interpretarea variabilelor in interiorul sirurilor

```
<?php
$juices = array("apple", "orange", "koolaid1" => "purple");
class people {
    public $john = "John Smith";
}

$people = new people();
echo "$people->john drank some $juices[o] juice.";
// John Smith drank some apple juice.
?>
```

Variabile tip string " "

- sintaxa **complexa** pentru interpretarea variabilelor in interiorul sirurilor **{ }**

```
<?php
$juice = "apple";

echo "He drank some juice made of $juices.";
// He drank some juice made of . //s character valid pentru variabile
echo "He drank some juice made of ${juice}s."
// He drank some juice made of apples. // {} arata unde se incheie
numele variabilei
?>
```

Variabile tip string " "

- sintaxa **complexa** pentru interpretarea variabilelor in interiorul sirurilor **{ }**

```
<?php
$juices = array(array("apple", "orange"), "koolaid1" => "purple");
class people {
    public $name = "John Smith";
}

$obj->values[3] = new people();
echo "$obj->values[3]->name drank some $juices[0][1] juice.";
// drank some juice.
echo "{ $obj->values[3]->name } drank some { $juices[0][1] } juice.";
// John Smith drank some apple juice.
?>
```

Variabile globale

Variabile globale

- Variabilele globale (predefinite)
 - accesibile script-urilor PHP prin conlucrarea cu server-ul
 - Exemple:
 - `$_SERVER` — Server and execution environment information
 - `$_GET` — HTTP GET variables
 - `$_POST` — HTTP POST variables
 - `$_FILES` — HTTP File Upload variables
 - `$_REQUEST` — HTTP Request variables
 - `$_SESSION` — Session variables
 - `$_ENV` — Environment variables
 - `$_COOKIE` — HTTP Cookies

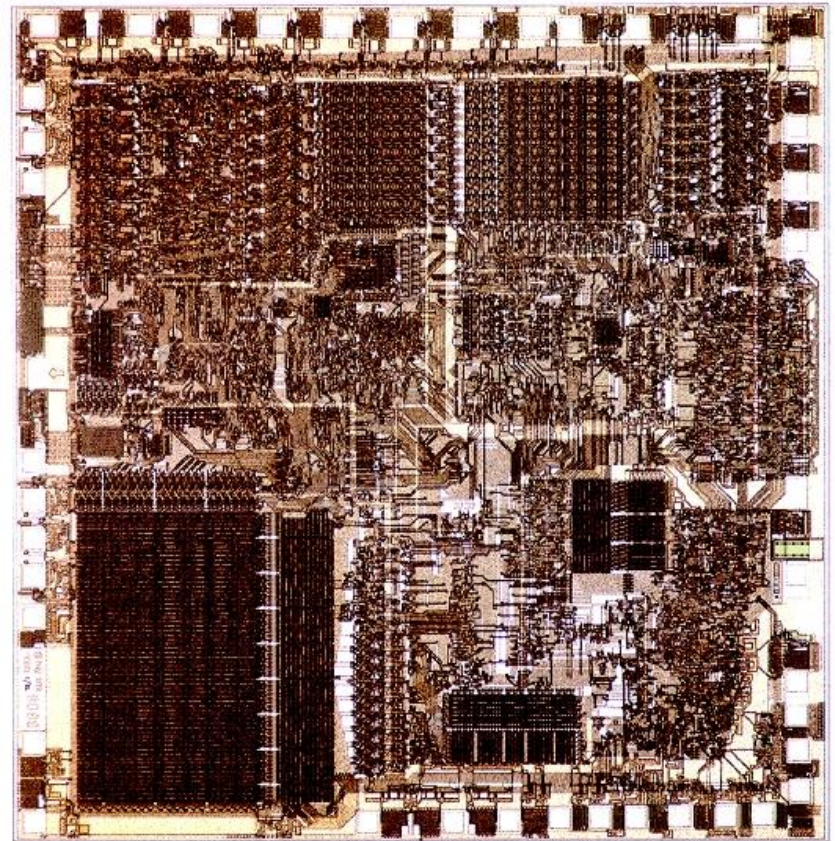
Interactiunea cu utilizatorul

- Datele introduse de utilizator in forme se regasesc (in functie de metoda aleasa pentru forma) in una din variabilele:
 - `$_POST` – method="post"
 - `$_GET` – method="get"
 - `$_REQUEST` – ambele metode
- variabilele sunt **matrici** cu **cheia** data de atributul **name** din forma HTML
 - `<input type="text" name="carti_cant" size="3" maxlength="3" />`
 - `$_POST['carti_cant']` contine valoarea introdusa

Structuri repetitive – tablouri

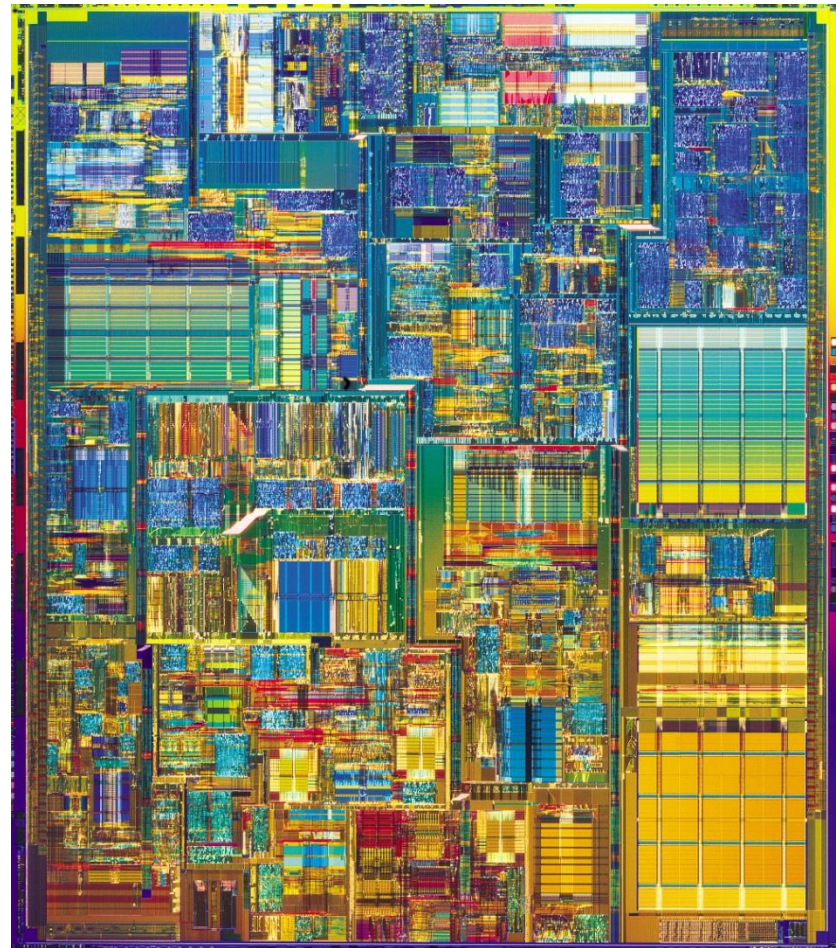
Impresionant?

- Intel® 8086
- 29.000 tranzistoare pe CPU
- 1978
- 1 MB date
- 4.7 MHz



Impresionant?

- Intel® Itanium® processors (Tukwila)
- 2009
- 2 miliarde tranzistoare pe CPU
- 16 EB date (16 G GB)
- > 3 GHz



Concepte

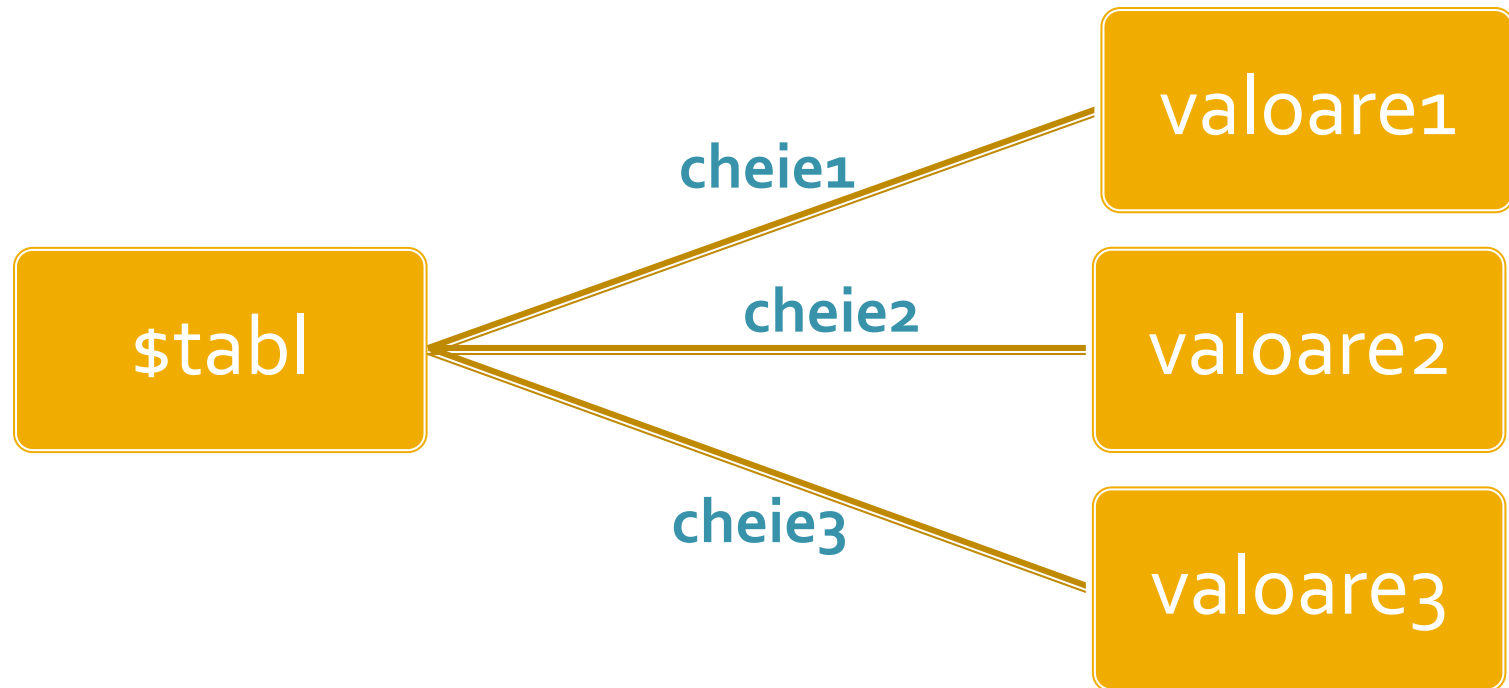
- Efectuare foarte rapida a unui numar **mic** de instructiuni, de **complexitate redusa**, repetate de un numar foarte mare de ori
- Programare: coborarea rationamentului la nivelul de **complexitate redusa**, cu obtinerea performantei prin structuri repetitive simple efectuate rapid.
- Operatii repetitive / date repetitive

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, ...

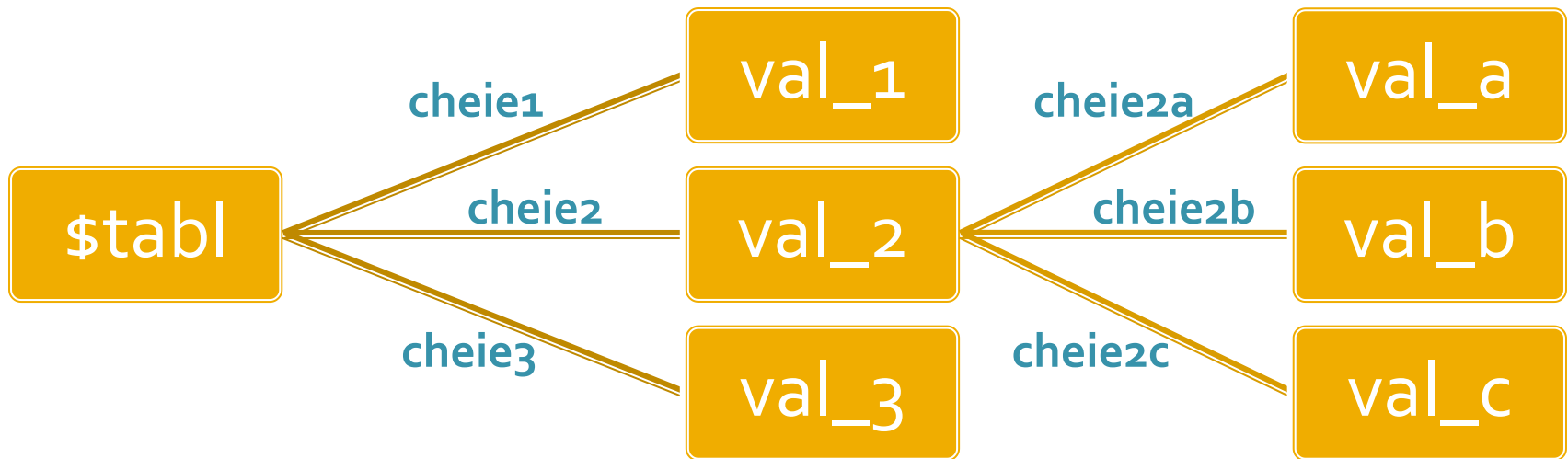
Tablou = Arbore

- `$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)`



Tablouri in PHP

```
$matr = array(1, 2, 3, 4, 5);
```

```
$matr[0]=1
```

```
$matr[1]=2
```

```
$matr[2]=3
```

```
$matr[3]=4
```

```
$matr[4]=5
```

```
$matr = array('a' => 1, 'b' => 2, 3, 4, 5);
```

```
$matr['a']=1
```

```
$matr['b']=2
```

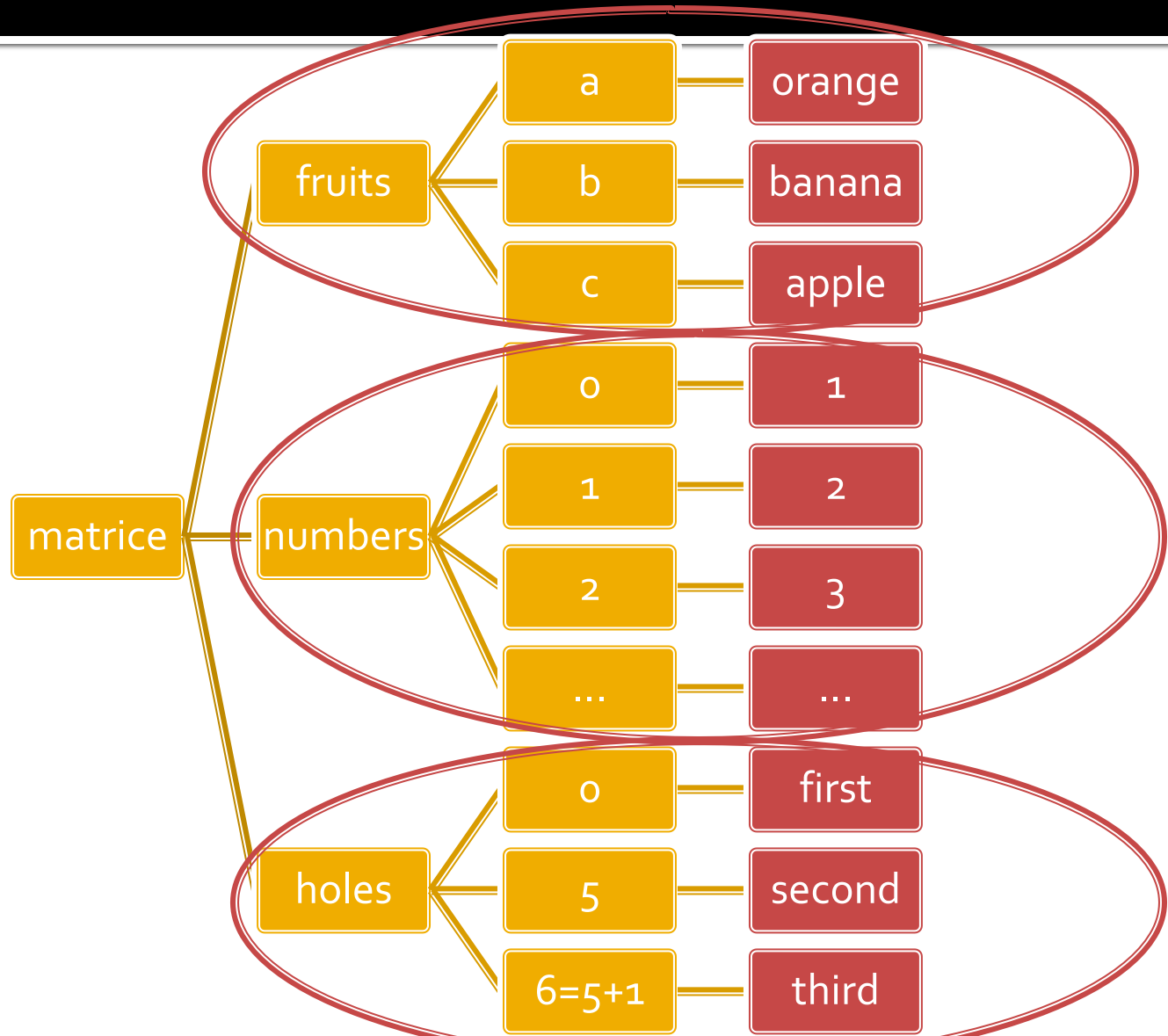
```
$matr[0]=3
```

```
$matr[1]=4
```

```
$matr[2]=5
```

```
$matrice= array (  
    "fruits" => array("a" => "orange", "b" => "banana", "c" => "apple"),  
    "numbers" => array(1, 2, 3, 4, 5, 6),  
    "holes" => array("first", 5 => "second", "third")  
);
```

Tablou = arbore



Afisarea tablourilor

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

```
$matr= array (  
"fruits" =>  
array("a" => "orange", "b" => "banana", "c" => "apple",  
"ultim"),  
"numbers" =>  
array(1, 2, 3, 4, 5, 6),  
"holes" =>  
array("first", 5 => "second", "third")  
);  
echo $matr;  
echo "<pre>";  
print_r ($matr);  
echo "</pre>";
```

```
Array  
  
Array  
(  
    [fruits] => Array  
        (  
            [a] => orange  
            [b] => banana  
            [c] => apple  
            [0] => ultim  
        )  
    [numbers] => Array  
        (  
            [0] => 1  
            [1] => 2  
            [2] => 3  
            [3] => 4  
            [4] => 5  
            [5] => 6  
        )  
    [holes] => Array  
        (  
            [0] => first  
            [5] => second  
            [6] => third  
        )  
)
```

Chei

- Chei numerice
 - implicite
 - similare celorlalte limbaje de programare
 - dificil de utilizat (trebuie retinuta valoarea logica a unei anumite chei numerice)
- Chei sir
 - claritate mai mare
 - eficienta numerica mai mica
 - tablourile au un index numeric intern, implicit ascuns, accesibil prin functii :
index => cheie => valoare

Elemente de control

- `for` – util dacă la definirea tablourilor sunt folosite cheile numerice implicite (numere întregi)
- `do ... while` și `while` se pot folosi împreună cu funcții specifice caracteristice tablourilor `next()`, `prev()`, `end()`, `reset()`, `current()`, `each()`
- `foreach` - elementul de control al iteratiilor cel mai potrivit pentru chei alfanumerice

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
```

Tablouri – functii utile

- `current ($matr)` – returneaza elementul indicat de indicele intern al tabloului (`~v[i]`)
- `next ($matr)` – incrementeaza indicele intern si returneaza valoarea stocata acolo (`~v[++i]`)
- `prev ($matr)` – decrementeaza indicele intern si returneaza valoarea stocata acolo (`~v[--i]`)
- `end($matr)` – muta indicele intern la ultimul element si returneaza valoarea stocata acolo (`~i=N-1;v[i]`)
- `reset($matr)` – muta indicele intern la primul element si returneaza valoarea stocata acolo (`~i=0;v[i]`)

Tablouri – functii utile

- `sort($matr)` – ordoneaza in ordine crescatoare a **valorilor** un tablou, cheile sunt sterse si recreate
 - `$fruits = array("lemon", "orange", "banana", "apple");`
`sort($fruits);`
 - `fruits[0] = apple, fruits[1] = banana, fruits[2] = lemon, fruits[3] = orange`
- `rsort($matr)` – similar, descrescator

Tablouri – functii utile

- `asort($matr)` ordoneaza in ordine crescatoare a **valorilor** un tablou, cheile sunt pastrate, inclusiv asocierea cheie => valoare
 - `$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");`
`asort($fruits);`
 - `c = apple, b = banana, d = lemon, a = orange`
- `arsort($matr)` – similar, descrescator

Tablouri – functii utile

- `ksort($matr)` ordoneaza in ordine crescatoare a **cheilor** un tablou, cheile sunt pastrate, inclusiv asocierea cheie => valoare
 - `$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");`
`ksort($fruits);`
 - a = orange, b = banana, c = apple , d = lemon
- `krsort($matr)` – similar, descrescator

Laborator 3

Laborator L3

- Sa se creeze un magazin simplu virtual care:
 - sa prezinte utilizatorului o lista de produse si preturi (constanta – maxim 5 produse)
 - sa preia de la acesta numarul de produse dorit
 - sa calculeze suma totala
 - sa adauge TVA 19%
 - sa prezinte un raport care sa contina:
 - total de plata
 - ora comenzii

Laborator L3 - continuare

- se creaza macar 3 pagini:
 - lista produse
 - formular comanda
 - rezultat
- forma paginilor:
 - tabel/CSS
- metoda
 - post
 - get

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	

Laborator – L3 - rezultat

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Pret
1	Carti	100
2	Caiete	50
3	Penare	150
4	Stilouri	125
5	Creioane	25

[Comanda](#)

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

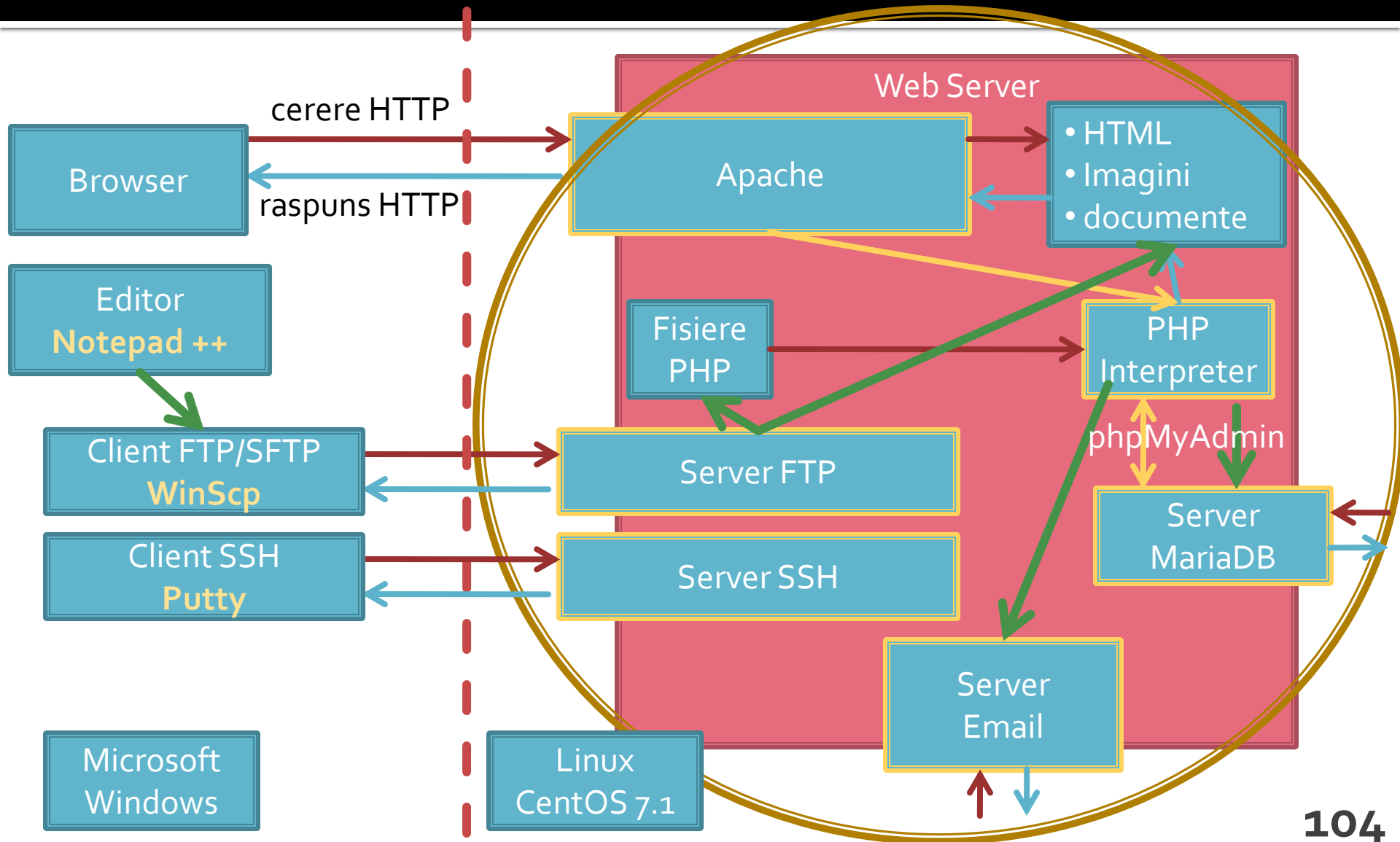
Rezultate comanda

Pret total (fara TVA): 350

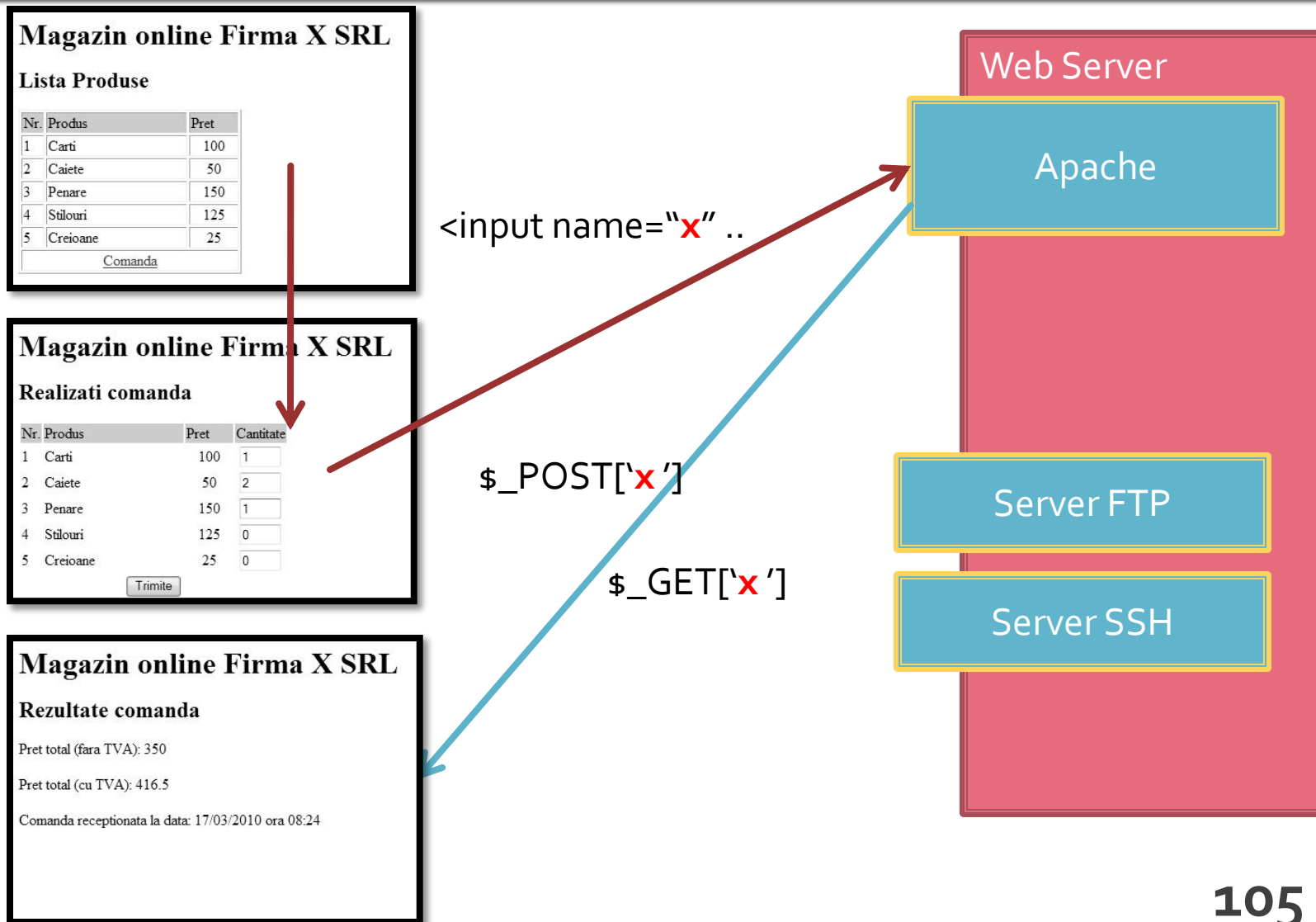
Pret total (cu TVA): 416.5

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

Utilizare LAMP

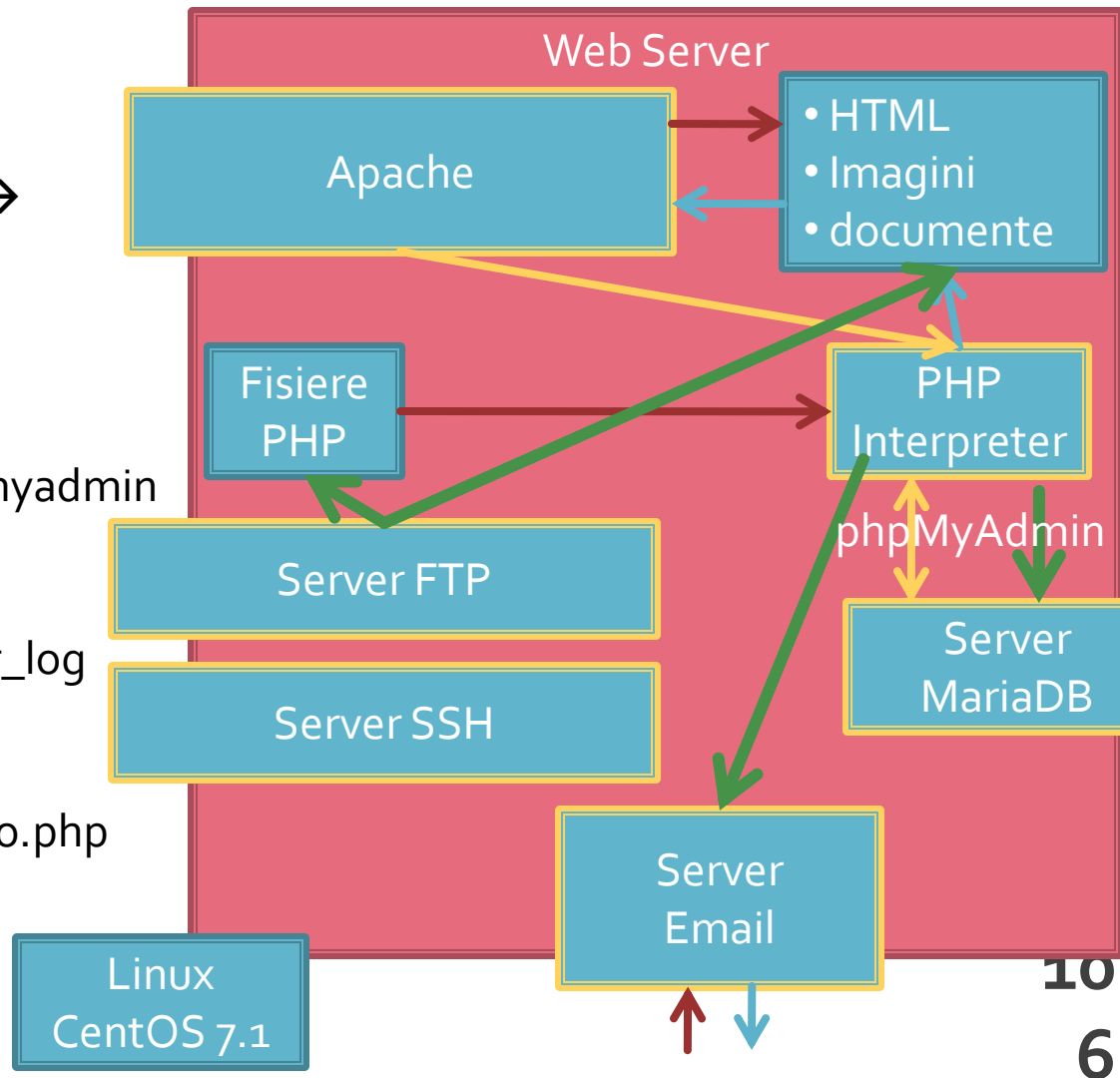


Utilizare LAMP



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



Client / Server

```
<input name="nume" ....>
```

```
echo $_POST['nume']; //ceva  
echo $_GET['nume']; //ceva  
echo $_REQUEST['nume']; //ceva
```

ceva

Trimite

get
post

Interpretor PHP primeste
\$_POST
\$_GET
\$_REQUEST

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