Curs 2

2020/2021

Programarea aplicațiilor web

PAW

- Programarea aplicațiilor web
  - An V RC
    - 1.5C/1L/1P

Program

- An V
  - Saptamana 1
    - Marti 18-20 Proiect
  - Saptamanile 2-8
    - Marti 16:30-18 Curs
    - Marti 18-20 Laborator
  - Saptamanile 9-14
    - Marti 16:30-18 Curs
    - Marti 18-20 Proiect

Orar

- www.etc.tuiasi.ro/orar : Curs 16 (->16:30)

FACULTATEA DE ELECTRONICA, TELECOMUNICATII SI TEHNOLOGIA INFORMATIEI

55RC

ETTI_

	1 8:00 - 8:50	2 9:00 - 9:50	3 10:00 - 10:50	4 11:00 - 11:50	5 12:00 - 12:50	6 13:00 - 13:50	7 14:00 - 14:50	8 15:00 - 15:50	9 16:00 - 16:50	10 17:00 - 17:50	11 18:00 - 18:50	12 19:00 - 19:50
L											Trifina L. TEFO (L) 1.08 Practica	
Ma									Damian R. PAW (C) Online_DaR		Damian R. PAW (L) Online_DaR Damian R. PAW (P) Online_DaR	
Mi									Sirbu A. POO (C) Online_SA		Casian-Botez I. Etic (C) Online_CBI	Casian-Botez I. Etic (S) Online_CBI
J								Scripcariu L. RCALSC (L) 2.13 TC (R)			Scripcariu L. RCALSC (C) Online_SL Alecsandrescu I. POO (L) 3.34 CONTI (L)	
V											Trifina L. TEFO (C) Online_TrL	
Sa												

http://www.etti.tuiasi.ro/orar aSc Timetables

Nota

- An V
  - 33% E: 40%
  - 66% Aplicatii
    - 33% L (0%)
    - 33% P (60%)

Acces

- ~~Personalizat~~

Date:

Grupa	5304 (2015/2016)
Specializarea	Tehnologii si sisteme de telecomunicatii
Marca	5184

Acceseaza ca acest student

Note obtinute

Disciplina	Tip	Data	Descriere	Nota	Puncte	Obs.
TW	Tehnologii Web					
	N	17/01/2014	Nota finala	10	-	
	A	17/01/2014	Colocviu Tehnologii Web 2013/2014	10	7.55	
	B	17/01/2014	Laborator Tehnologii Web 2013/2014	9	-	
	D	17/01/2014	Tema Tehnologii Web 2013/2014	9	-	

Nume

Email

Cod de verificare

344bd9f

Trimite

Online

- acces la **examene** necesita **parola** primita prin email

Online

- acces email/parola

Parola

- primita prin email

Important message from RF-OPTO Inbox ×

Radu-Florin Damian
to me, POPESCU

Romanian > English Translate message

Laboratorul de Microunde si Optoelectronica
Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei
Universitatea Tehnica "Gh. Asachi" Iasi

In atentia: POPESCU GOPO ION

Parola pentru a accesa examenele pe server-ul **rf-opto** este
Parola: [redacted]

Identificati-va pe server, cu parola, cat mai rapid, pentru confirmare.

Memorati acest mesaj intr-un loc sigur, pentru utilizare ulterioara

---

Attention: POPESCU GOPO ION

The password to access the exams on the **rf-opto** server is
Password: [redacted]

Login to the server, with this password, as soon as possible, for confirmation.

Save this message in a safe place for later use

Reply Reply all Forward

Subject	Correspondents
Important message from RF-OPTO	POPESCU GOPO ION
Validation of MDCR exam from 02/05/2020	[redacted]
[redacted]	[redacted]

From Me <rdamian@etti.tuiasi.ro>
Subject **Important message from RF-OPTO**
To [redacted]
Cc Me <rdamian@etti.tuiasi.ro>

Laboratorul de Microunde si Optoelectronica
Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei
Universitatea Tehnica "Gh. Asachi" Iasi

In atentia: POPESCU GOPO ION

Parola pentru a accesa examenele pe server-ul **rf-opto** este
Parola: [redacted]

Identificati-va pe server, cu parola, cat mai rapid, pentru confirmare.

Memorati acest mesaj intr-un loc sigur, pentru utilizare ulterioara

---

Attention: POPESCU GOPO ION

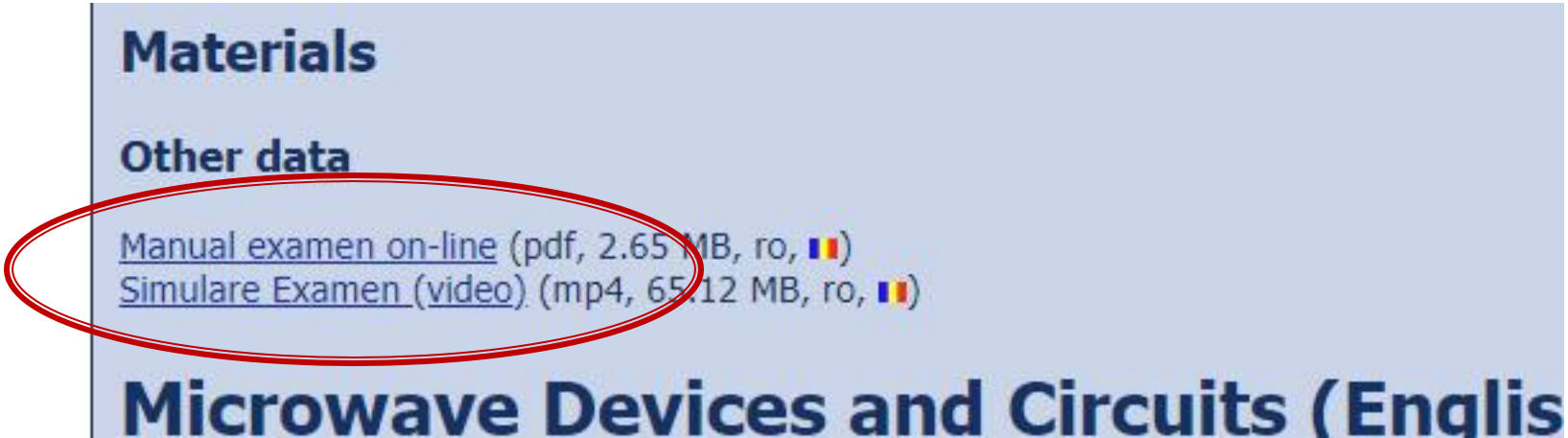
The password to access the exams on the **rf-opto** server is
Password: [redacted]

Login to the server, with this password, as soon as possible, for confirmation.

Save this message in a safe place for later use

Manual examen online

- Aplicatia de examen online utilizata intens la:
  - curs (prezenta)
  - laborator
  - proiect
  - examen



Corectii

PHP

- Hypertext PreProcessor - acronim recursiv
  - initial – Personal Home Page / Form Interpreter
  - 1995 – 1.0
- versiune curenta: 7.4.16
  - 2021-03-04
- versiune curenta: 8.0.3
  - 2021-03-04
- ~~versiune curenta: 5.6.40~~ **End of life**
  - 2019-01-10

Server Side Scripting

	2010 1 Jan	2011 1 Jan	2012 1 Jan	2013 1 Jan	2014 1 Jan	2015 1 Jan
PHP	72.5%	75.3%	77.3%	78.7%	81.6%	80.6%
ASP.NET	24.4%	23.4%	21.7%	20.2%	18.2%	16.7%
Java	4.0%	3.8%	4.0%	4.1%	2.7%	2.8%
ColdFusion		1.3%	1.2%	1.1%	0.8%	0.7%
Perl		1.1%	1.0%	0.8%	0.6%	0.5%
Ruby	0.5%	0.5%	0.6%	0.5%	0.4%	0.9%
Python	0.3%	1.0%	1.3%	1.5%	1.7%	1.6%
JavaScript			<0.1%	<0.1%	0.1%	0.1%

Server Side Scripting

	2016 1 Jan	2017 1 Jan	2018 1 Jan	2019 1 Jan	2020 1 Jan	2021 1 Jan
PHP	80.0%	80.0%	80.2%	78.9%	78.9%	79.1%
ASP.NET	15.6%	14.8%	13.5%	11.8%	10.6%	9.3%
Ruby	1.1%	1.3%	1.6%	2.4%	3.0%	4.3%
Java	3.1%	3.3%	3.4%	4.0%	3.7%	3.2%
Scala	0.2%	0.3%	0.5%	1.2%	1.6%	1.8%
static files	1.5%	1.5%	1.6%	2.1%	1.8%	1.6%
Python	1.7%	1.6%	1.3%	1.1%	1.3%	1.4%
JavaScript	0.2%	0.3%	0.4%	0.7%	0.8%	1.2%

TIOBE Programming Community Index for March 2020

Position Mar 2021	Position Mar 2020	Delta in Position	Programming Language	Ratings Mar 2021	Delta Mar 2020
1	2	↑	C	15.33%	-1.00%
2	1	↓	Java	10.45%	-7.33%
3	3	=	Python	10.31%	0.20%
4	4	=	C++	6.52%	-0.27%
5	5	=	C#	4.97%	-0.35%
6	6	=	Visual Basic	4.85%	-0.40%
7	7	=	JavaScript	2.11%	0.06%
8	8	=	PHP	2.07%	0.05%
9	12	↑	Assembly language	1.97%	0.72%
10	9	↓	SQL	1.87%	0.03%

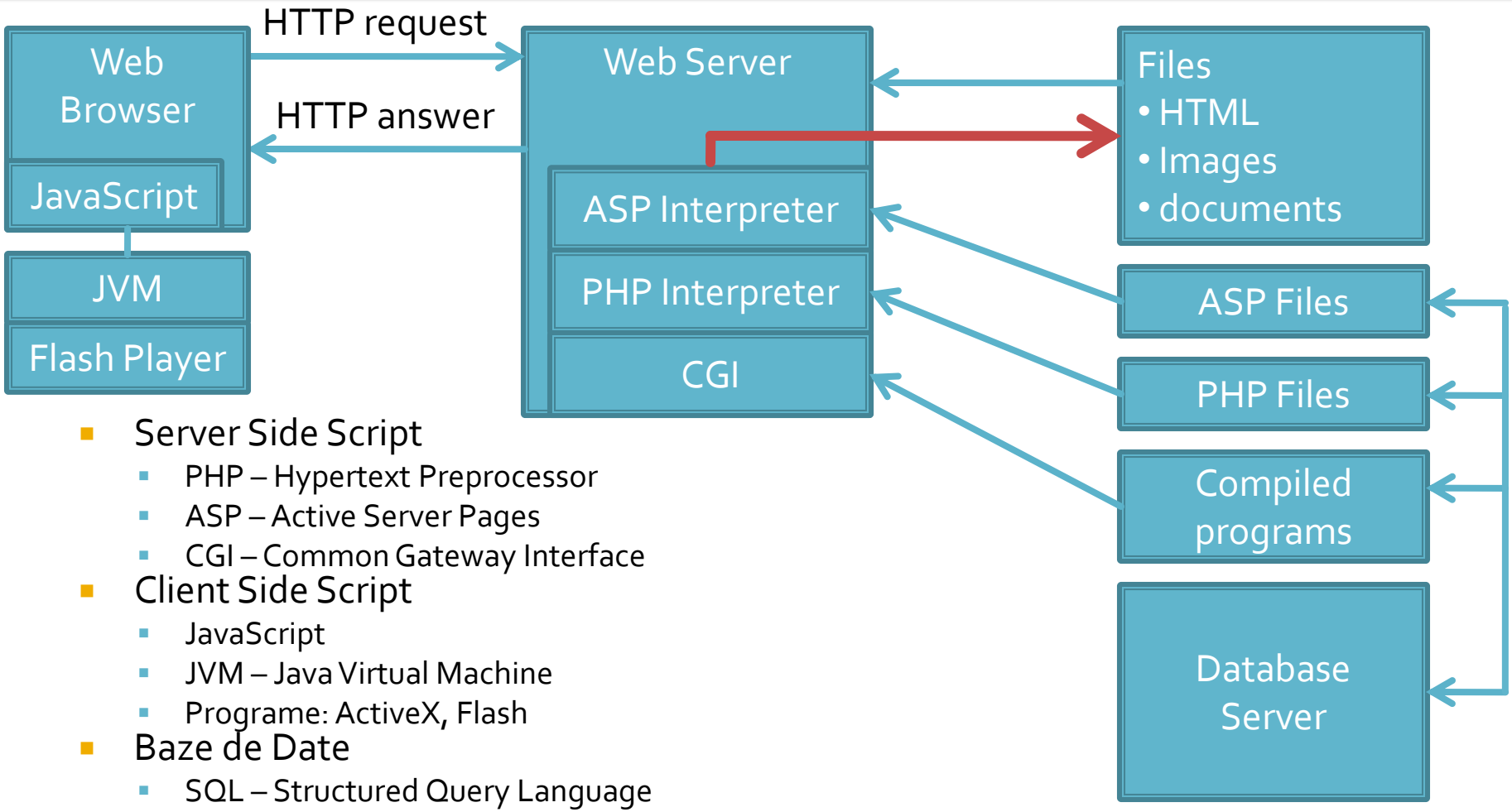
Hypertext PreProcessor

PHP

PHP

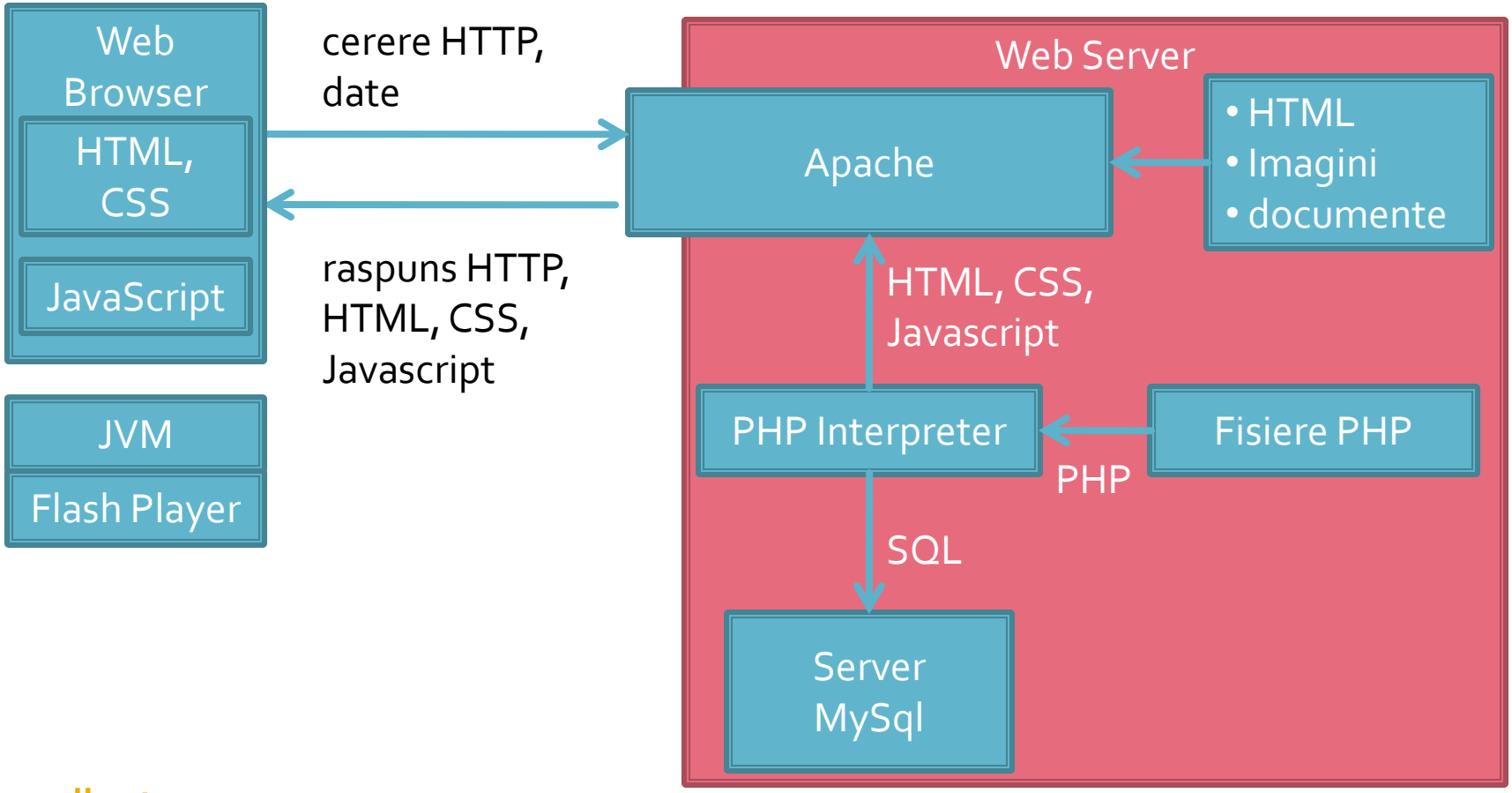
- Hypertext PreProcessor - acronim recursiv
  - initial – Personal Home Page / Form Interpreter
  - 1995 – 1.0
- versiune curenta: 5.6.40
  - 2019-01-10
- versiune curenta: 7.4.7
  - 2020-06-11
- limbaj de scripting de uz general,
- rulare pe server (server-side scripting)
- open source

Web server Technology



- Server Side Script
  - PHP – Hypertext Preprocessor
  - ASP – Active Server Pages
  - CGI – Common Gateway Interface
- Client Side Script
  - JavaScript
  - JVM – Java Virtual Machine
  - Programe: ActiveX, Flash
- Baze de Date
  - SQL – Structured Query Language
  - MySql – open Source
  - Microsoft SQL Server
  - Oracle

Client/Server Scripting

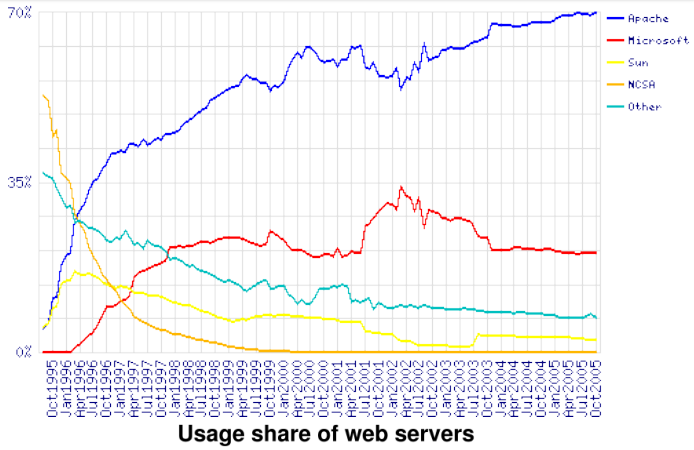


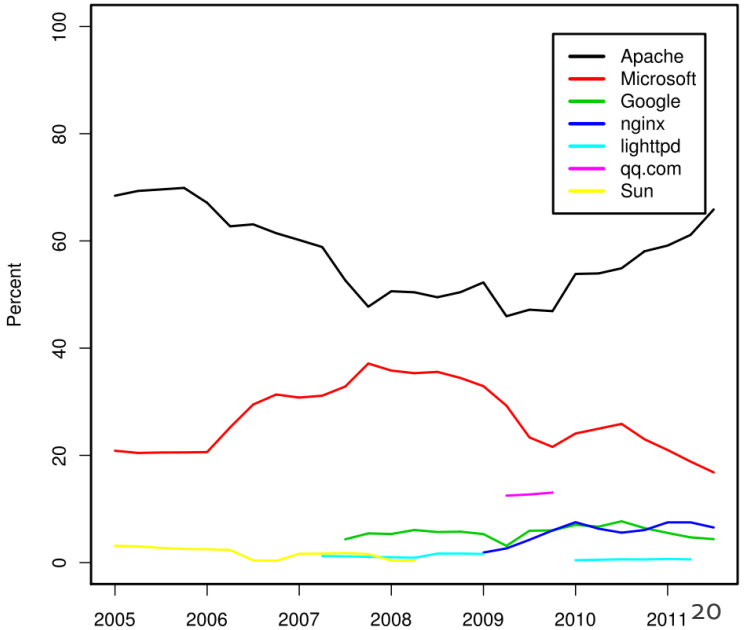
client side scripting

server side scripting

Tehnologia server-elor Web

- PHP – Hypertext Preprocessor
  - initially – Personal Home Page
  - open source
  - C++
  - Apache
- ASP – Active Server Pages
  - Microsoft
  - VBasic
  - IIS
- Java/JavaScript
  - Sun
  - Java Virtual Machine





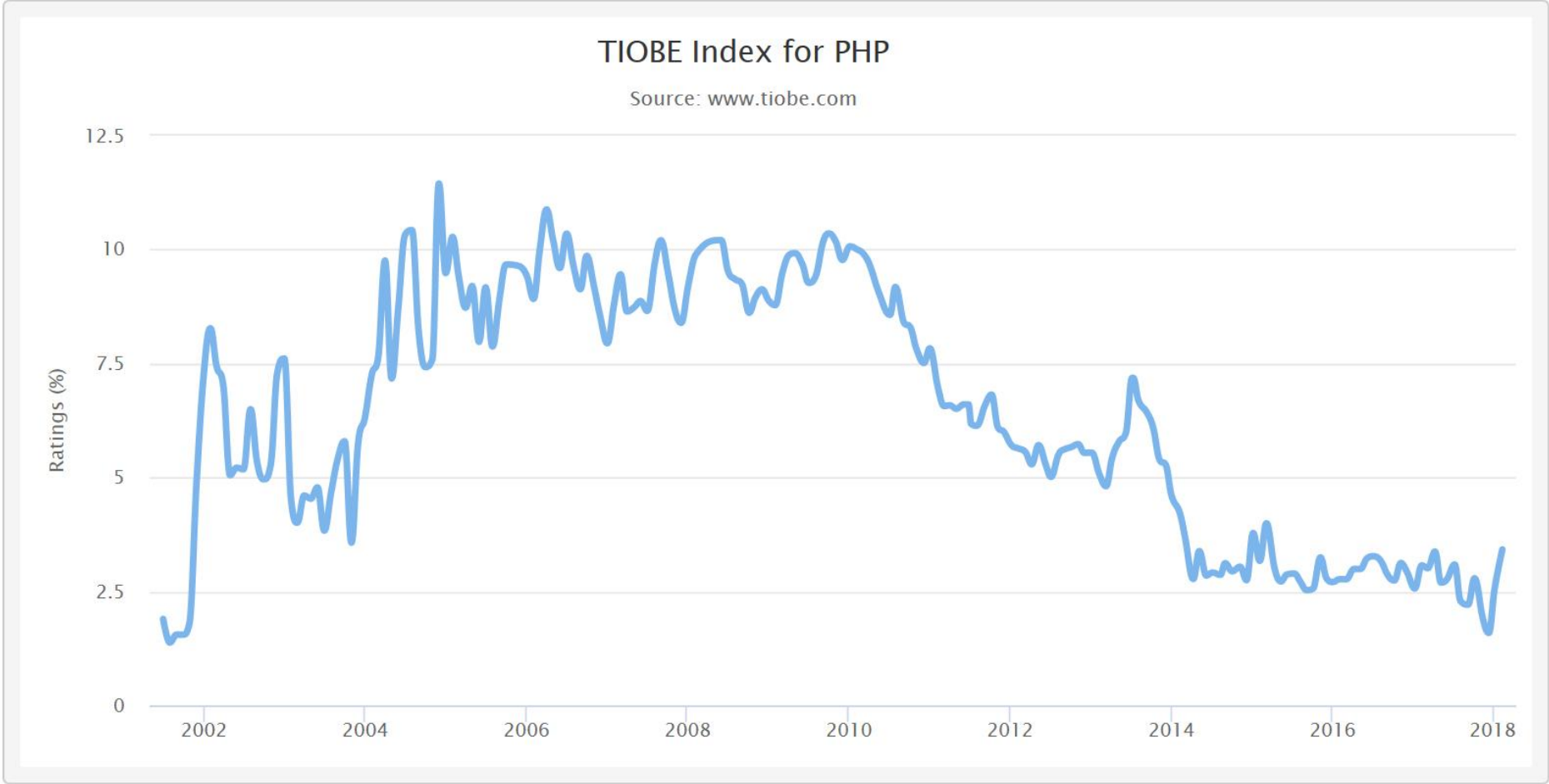
TIOBE Programming Community Index for February 2018

Position Feb 2018	Position Feb 2017	Delta in Position	Programming Language	Ratings Feb 2018	Delta Feb 2017
1	1	=	Java	14.99%	-1.69%
2	2	=	C	11.86%	3.41%
3	3	=	C++	5.73%	0.30%
4	5	↑	Python	5.17%	1.12%
5	4	↓	C#	4.45%	-0.45%
6	8	↑	Visual Basic .NET	4.07%	1.25%
7	6	↓	PHP	3.42%	0.35%
8	7	↓	JavaScript	3.17%	0.29%
9	9	=	Delphi/Object Pascal	2.59%	0.11%
10	11	↑	Ruby	2.53%	0.38%

TIOBE Programming Community Index for March 2010

Position Mar 2010	Position Mar 2009	Delta in Position	Programming Language	Ratings Mar 2010	Delta Mar 2009	Status
1	1	=	Java	17.509%	-2.29%	A
2	2	=	C	17.279%	+1.42%	A
3	4	↑	PHP	9.908%	+0.42%	A
4	3	↓	C++	9.610%	-0.75%	A
5	5	=	(Visual) Basic	6.574%	-1.71%	A
6	7	↑	C#	4.264%	-0.06%	A
7	6	↓	Python	4.230%	-0.95%	A
8	9	↑	Perl	3.821%	+0.40%	A
9	10	↑	Delphi	2.684%	-0.03%	A
10	8	↓↓	JavaScript	2.651%	-0.96%	A

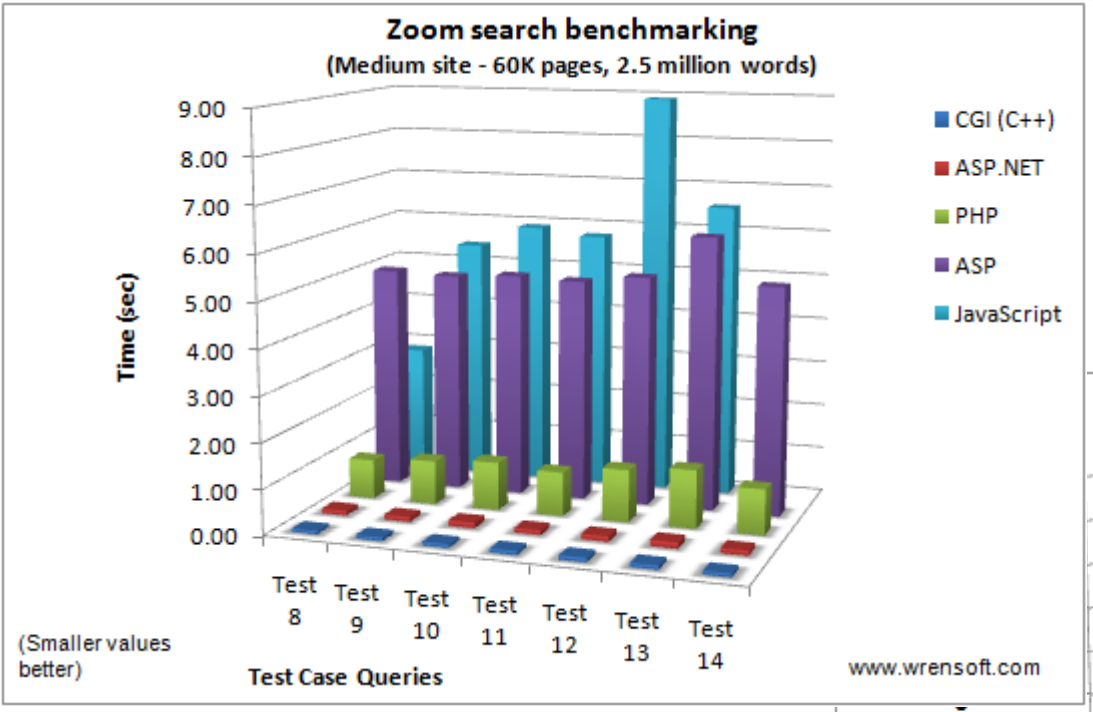
Evolutie PHP

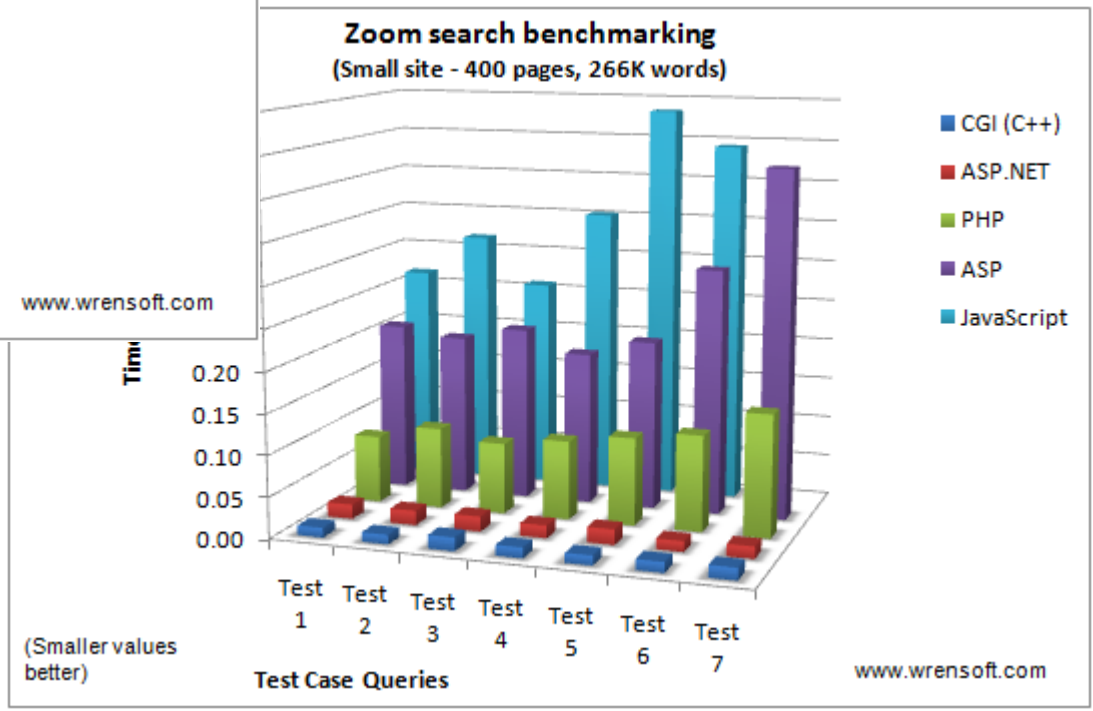


Avantaje PHP

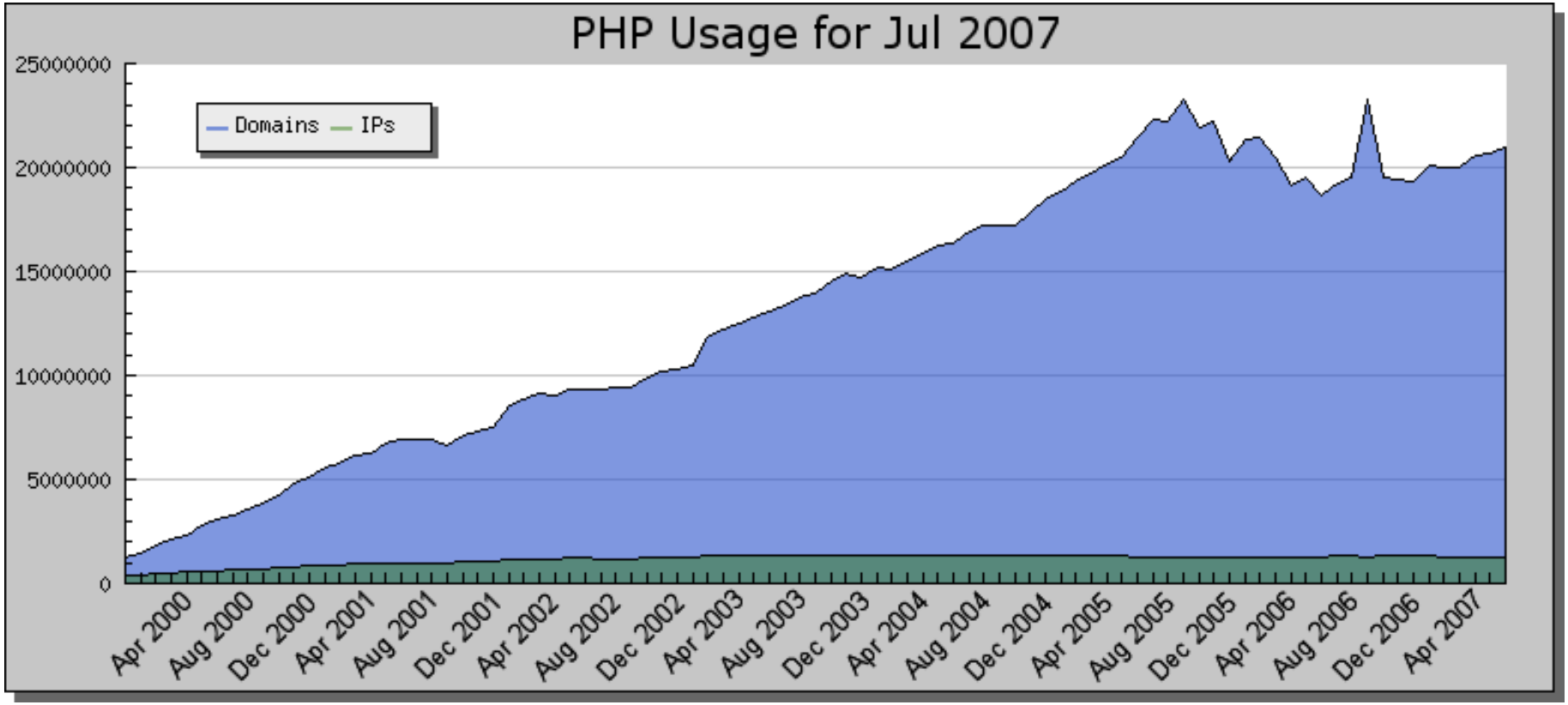
- Performanta ridicata
- Interfata cu multe sisteme de baze de date
- Costuri reduse
- Biblioteci incluse pentru majoritatea operatiunilor uzuale in aplicatii web
- Usurinta in invatare (C)
- Portabilitate
  - Disponibilitate a codului sursa
- Exemple disponibile in comunitate
- Suport disponibil

Performanta

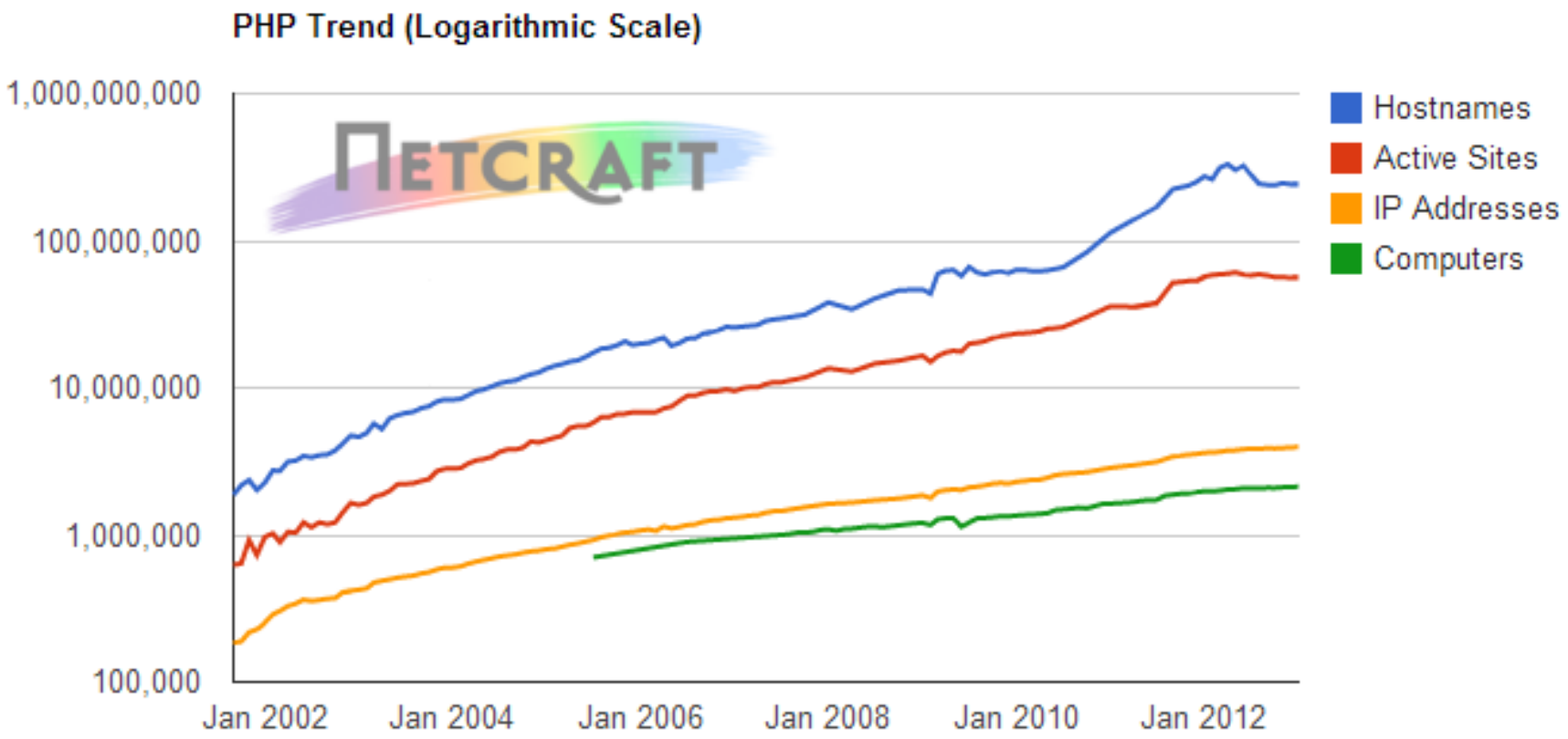


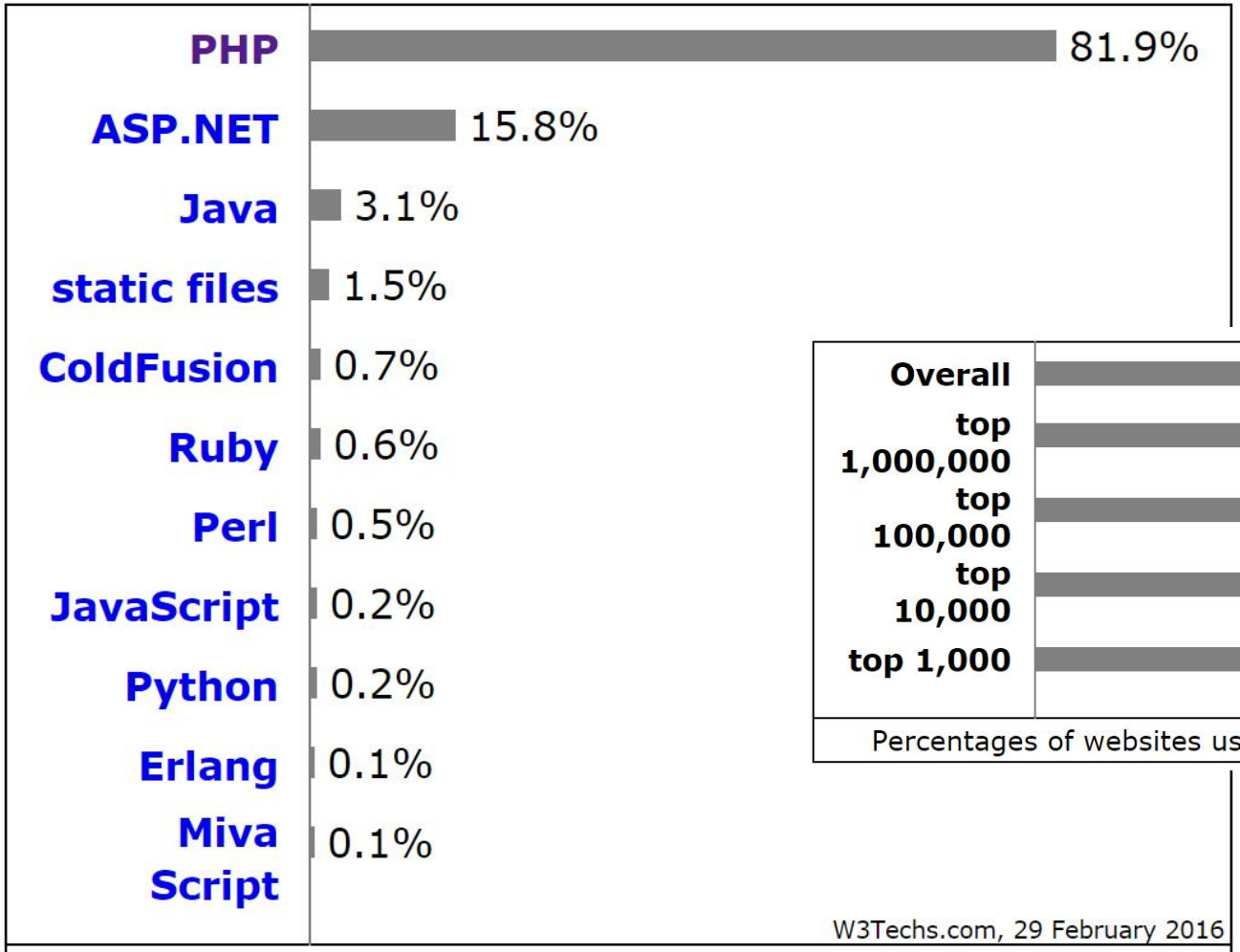


Raspandire

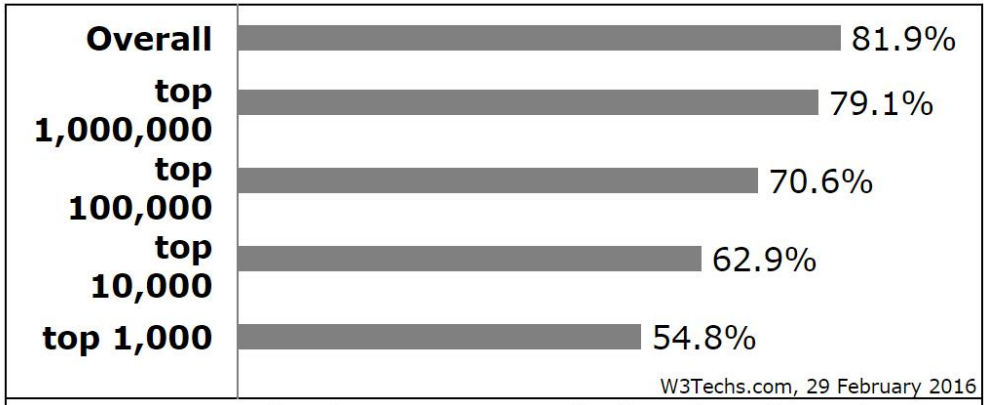

PHP Usage for Jul 2007
Domains
IPs
25000000
20000000
15000000
10000000
5000000
0
Apr 2000
Aug 2000
Dec 2000
Apr 2001
Aug 2001
Dec 2001
Apr 2002
Aug 2002
Dec 2002
Apr 2003
Aug 2003
Dec 2003
Apr 2004
Aug 2004
Dec 2004
Apr 2005
Aug 2005
Dec 2005
Apr 2006
Aug 2006
Dec 2006
Apr 2007

Raspandire

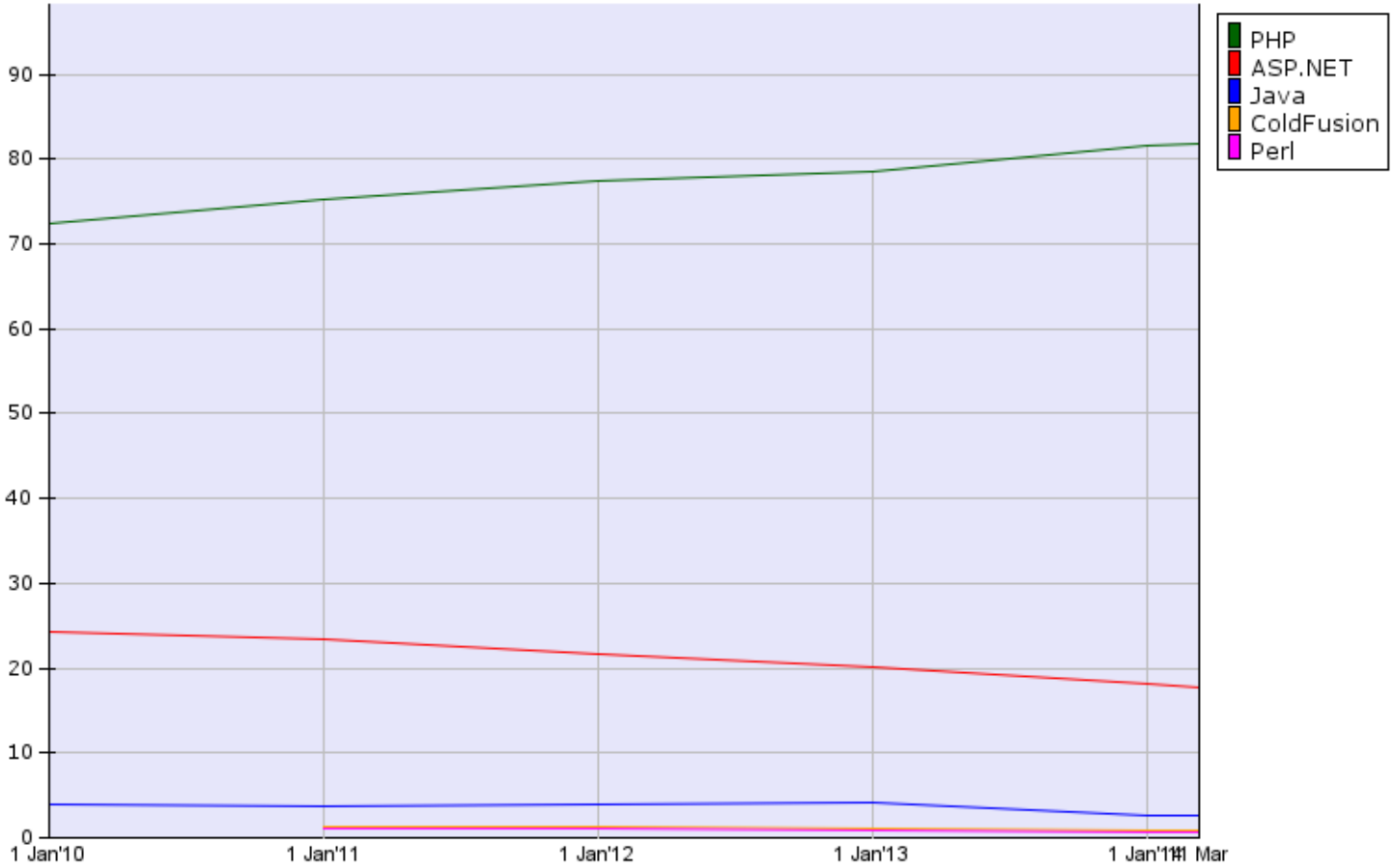

PHP Trend (Logarithmic Scale)
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
Jan 2002
Jan 2004
Jan 2006
Jan 2008
Jan 2010
Jan 2012
NETCRAFT
Hostnames
Active Sites
IP Addresses
Computers



Percentages of websites using various server-side programming languages



Percentages of websites using PHP broken down by ranking



Usage of server-side programming languages for websites, 11 Mar 2014, W3Techs.com

Server Side Scripting

	2010 1 Jan	2011 1 Jan	2012 1 Jan	2013 1 Jan	2014 1 Jan	2014 11 Mar
PHP	72.5%	75.3%	77.3%	78.7%	81.6%	81.9%
ASP.NET	24.4%	23.4%	21.7%	20.2%	18.2%	17.8%
Java	4.0%	3.8%	4.0%	4.1%	2.7%	2.7%
ColdFusion		1.3%	1.2%	1.1%	0.8%	0.8%
Perl		1.1%	1.0%	0.8%	0.6%	0.6%
Ruby	0.5%	0.5%	0.6%	0.5%	0.4%	0.5%
Python	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%
JavaScript			<0.1%	<0.1%	0.1%	0.1%

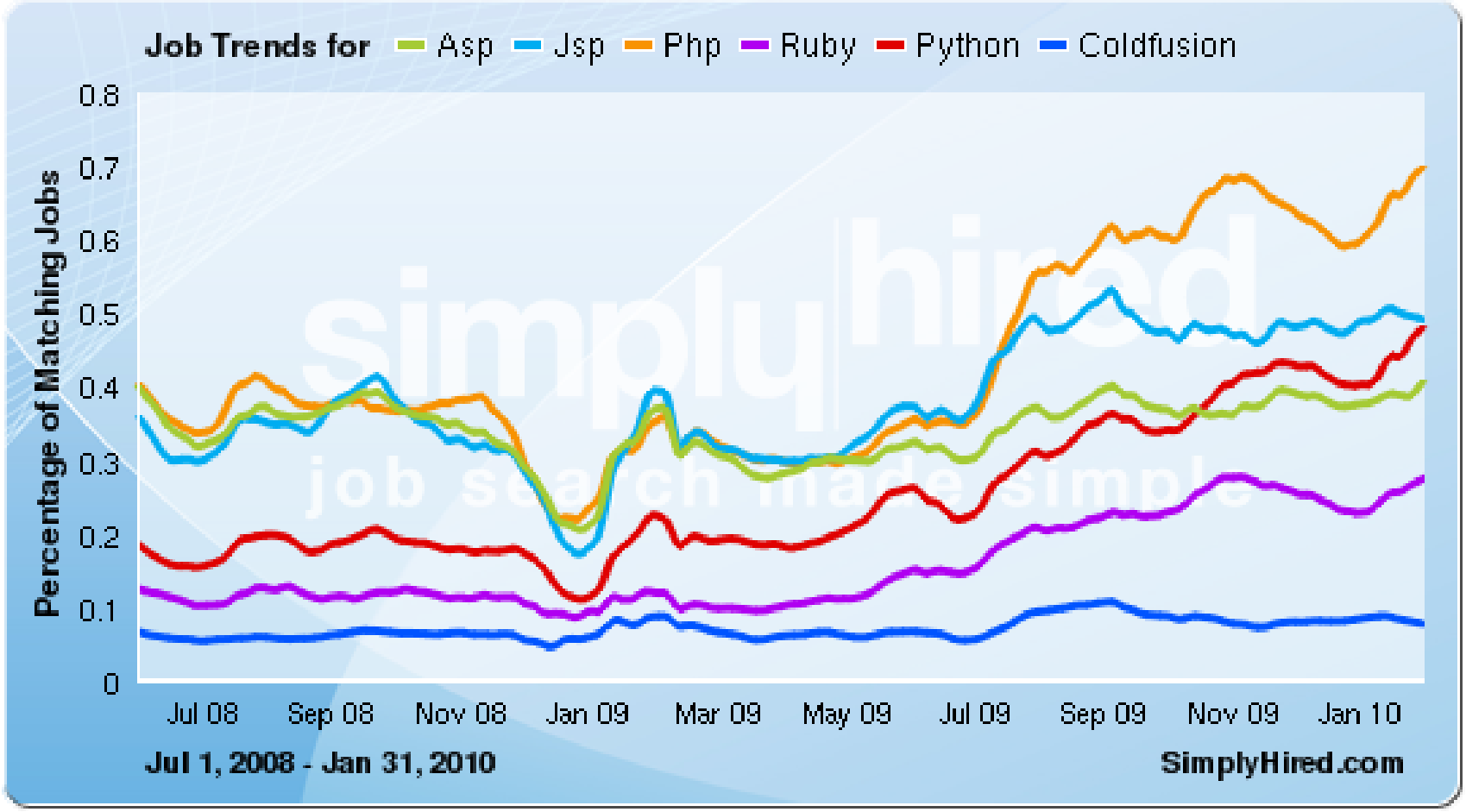
Server Side Scripting

	2015 1 Feb	2015 1 Jul	2016 1 Jan	2016 15 Feb
PHP	82.0%	81.8%	81.7%	81.8%
ASP.NET	17.0%	16.9%	16.0%	15.9%
Java	2.8%	3.0%	3.0%	3.0%
static files			1.6%	1.6%
ColdFusion	0.7%	0.7%	0.7%	0.7%
Ruby	0.6%	0.6%	0.6%	0.6%
Perl	0.5%	0.5%	0.5%	0.5%
Python	0.2%	0.2%	0.2%	0.2%

Sites

- Facebook.com
- Wikipedia.org
- Qq.com
- Taobao.com
- Sina.com.cn
- Wordpress.com
- Vk.com
- Weibo.com
- Babylon.com
- Mail.ru

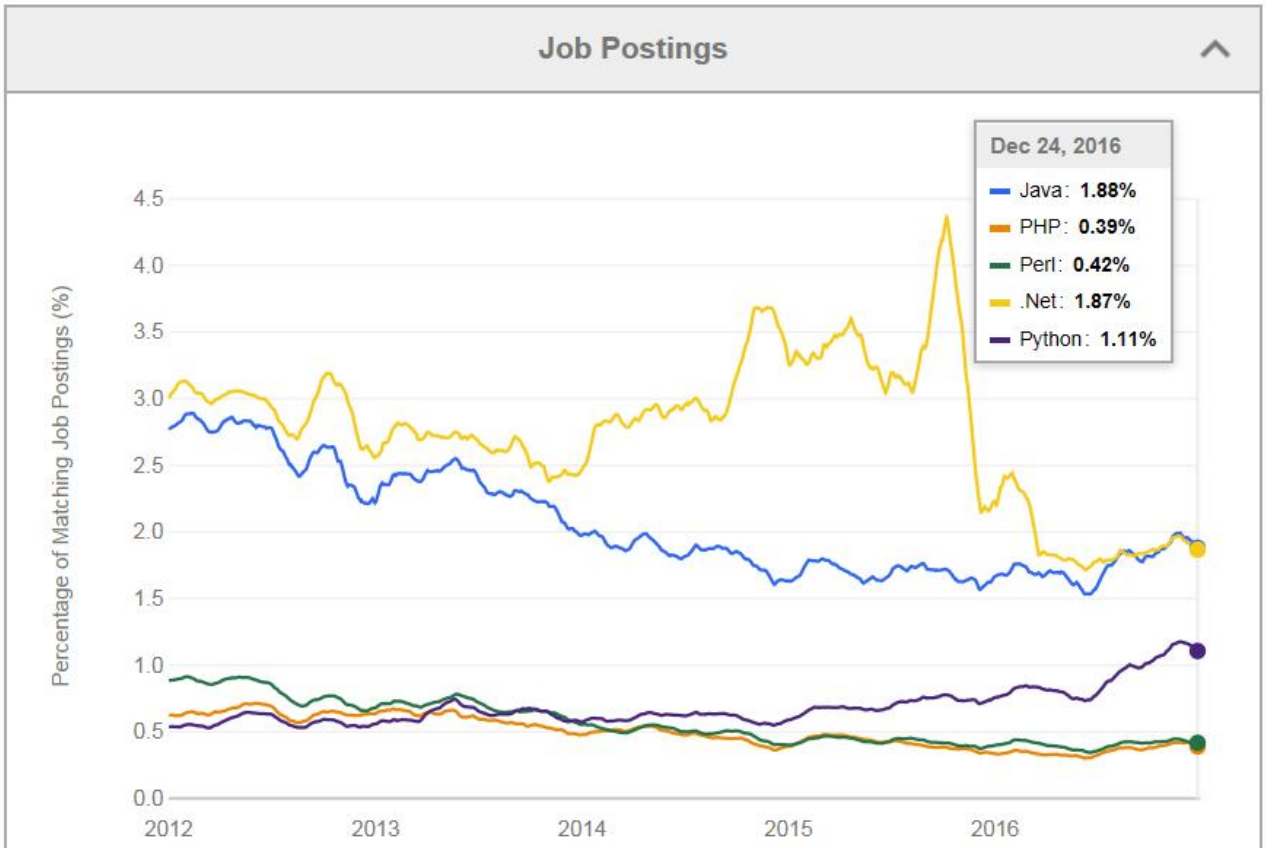
Angajare

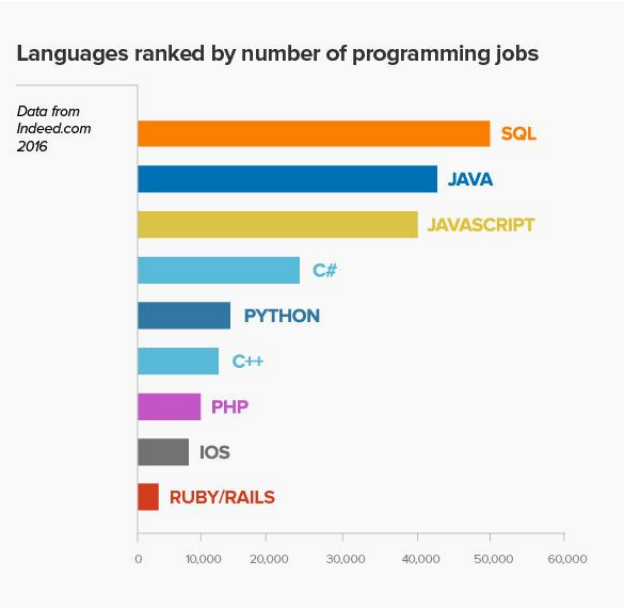

Job Trends for
Asp
Jsp
Php
Ruby
Python
Coldfusion
Percentage of Matching Jobs
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Jul 08
Sep 08
Nov 08
Jan 09
Mar 09
May 09
Jul 09
Sep 09
Nov 09
Jan 10
Jul 1, 2008 - Jan 31, 2010
SimplyHired.com

Angajare

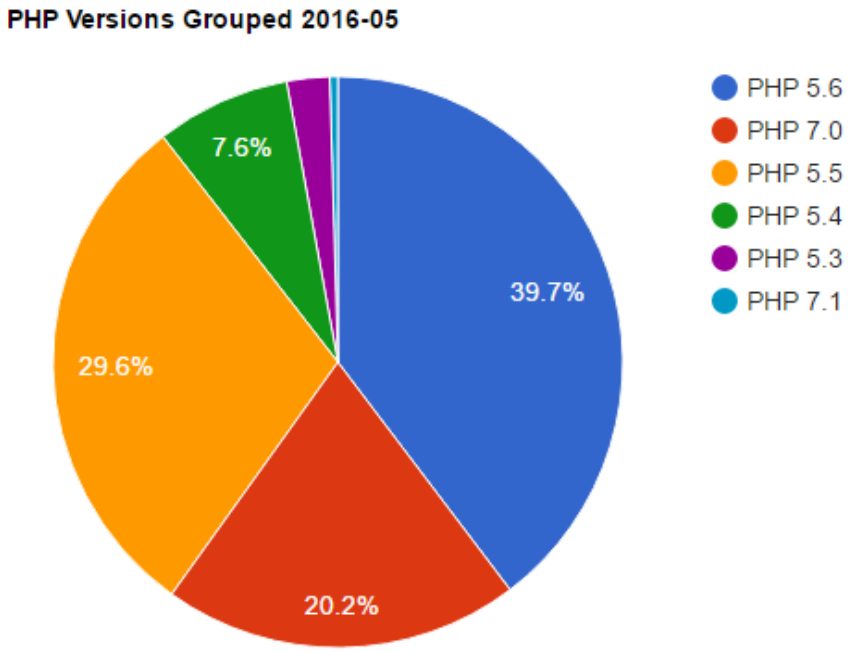
Java, PHP, Perl, .Net, and Python Job Trends

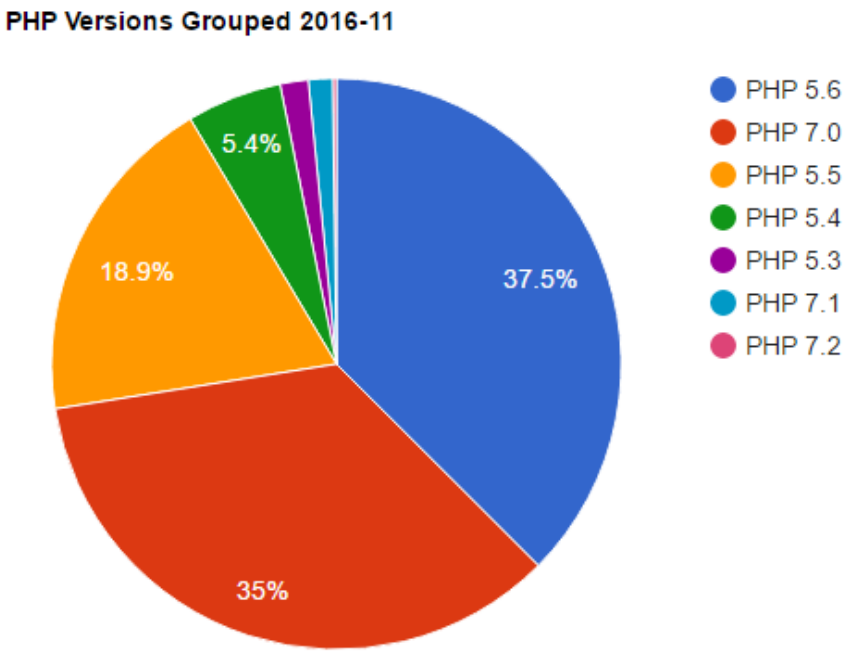
Java × | PHP × | Perl × | .Net × | Python × | + Add Term | Find Trends



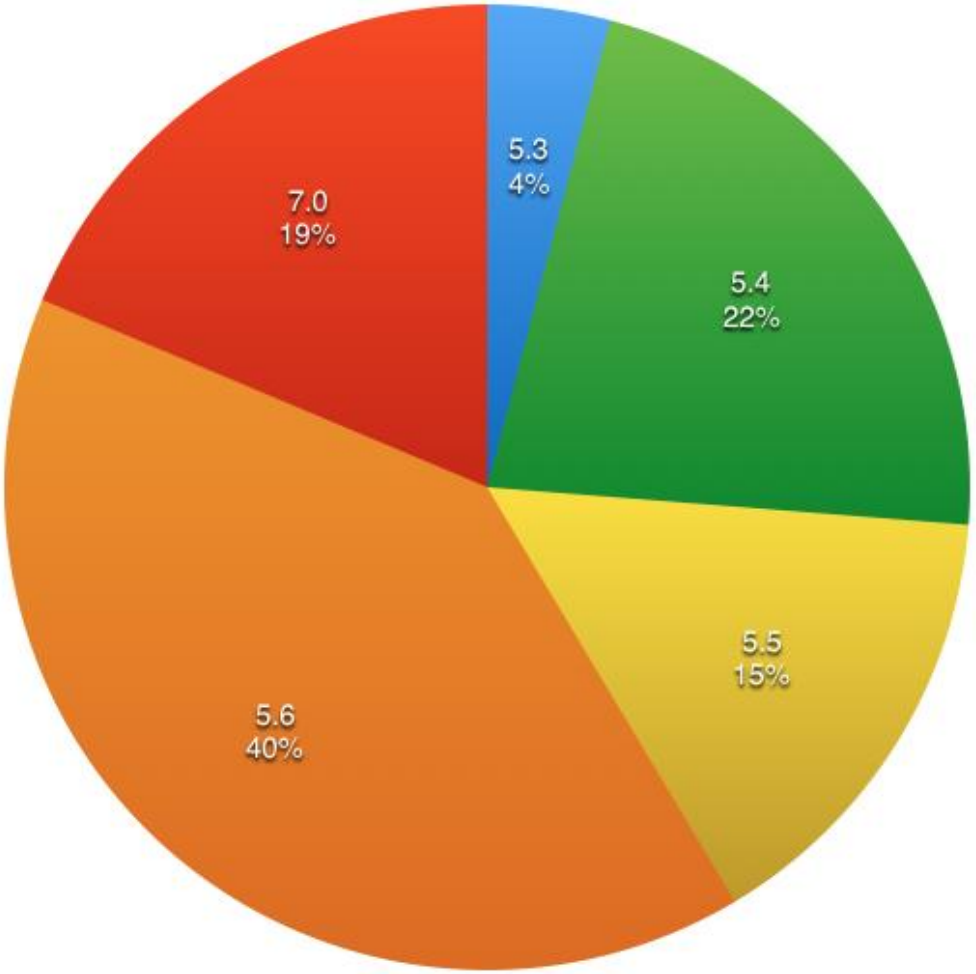


PHP 7, Sisteme





PHP 7, Aplicatii



Ex. Server

Server Information

Server Information

Item	Detail
Hosting Package	5G
Server Name	server72
cPanel Version	62.0 (build 8)
Apache Version	2.4.25
PHP Version	5.6.30
MySQL Version	10.0.27-MariaDB-cll-lve
Architecture	x86_64
Operating System	linux
Shared IP Address	188.212.252.74
Path to Sendmail	/usr/sbin/sendmail
Path to Perl	/usr/bin/perl
Perl Version	5.16.3
Kernel Version	3.10.0-427.18.2.lve1.4.27.el7.x86_64

Server Information

Item	Detail
Hosting Package	personal
Server Name	server11
cPanel Version	60.0 (build 37)
Apache Version	2.2.27
PHP Version	5.4.28
MySQL Version	5.5.52-cll
Architecture	x86_64
Operating System	linux
Shared IP Address	88.198.167.40
Path to Sendmail	/usr/sbin/sendmail
Path to Perl	/usr/bin/perl
Perl Version	5.10.1
Kernel Version	2.6.32-458.23.2.lve1.2.48.el6.x86_64

Ex. Server

System Information

System Item	Details	Status
Server Load	12.708984 (32 CPUs)	✓
Memory Used	60.9% (160,698,628 of 263,854,760)	✓
Swap Used	10.47% (7,028,300 of 67,108,860)	✓

Disk Information

Device	Mount Point	Usage	
/home	/home	79% (867,228,844 of 1,152,542,444)	✓
/	/	41% (153,355,872 of 393,757,580)	✓
/tmp	/tmp	1% (52,332 of 3,997,376)	✓
/var/tmp	/var/tmp	1% (52,332 of 3,997,376)	✓
/boot	/boot	61% (277,346 of 487,634)	✓

Server Information

Processor Information

Total processors: 32

Processor #1

Vendor
GenuineIntel

Name
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz

Speed
2600.343 MHz

Cache
20480 KB

Processor #2

Vendor
GenuineIntel

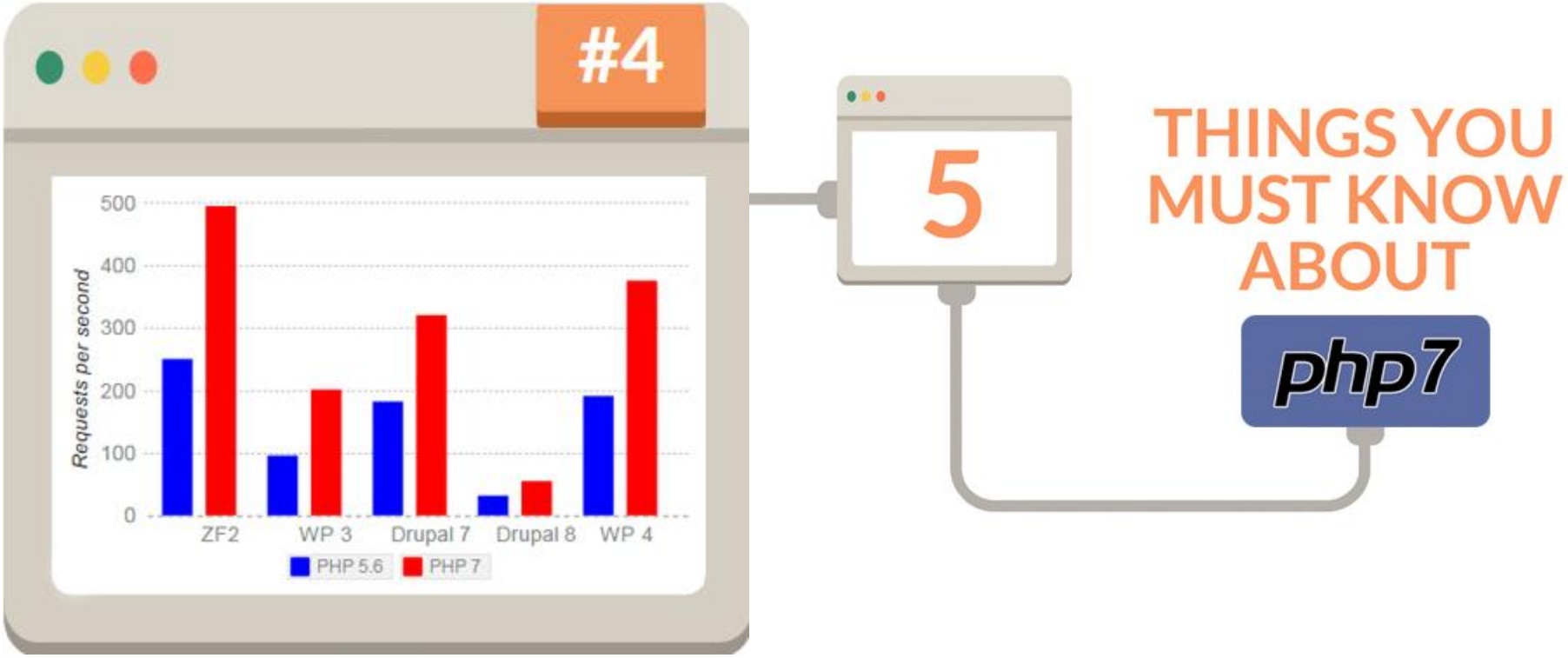
Name
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz

Speed
2599.968 MHz

Cache
20480 KB

Processor #3

PHP 7, performanta



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'];?> </td>
540         <td><?php echo $row['user_creat'];?> </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">
```

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 – impicit, 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

HTML + PHP

E:\Documents\Pagini Web\Safir dep\lista_lot.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

logfile.php | lista_lot.php

```
539  <?php }?>
540
541  <div id="maincontent">
542  <p><span class="title">Adauga</span><br/>
543    <a href="control_lot.php?id=0">Lot nou</a></p>
544  <p class="title">Loturi active</p>
545  <?php
546  $query = "SELECT l.*, c.`nume_user` AS `user_creat`
547          FROM `lot` AS l
548          LEFT JOIN `users` AS c ON (l.`User`=c.`id_user`)
549          WHERE l.`Activ` <> 0 ORDER BY l.`ID_LOT` DESC";
550  $result = mysql_query($query);
551  $total=0;
552  if ($result && (mysql_num_rows($result) > 0))
553      {
554      $total=mysql_num_rows($result);
555      $row = mysql_fetch_assoc($result);
556      }
557  if ($total>0)
558  {?>
559  <table align="center">
560    <tr class="lista_titlu">
561      <td align="center">Nr.</td>
562        <td>Numar</td>
563        <td>Data</td>
564        <td>Nr. pui</td>
565        <td>Gr. pui</td>
566        <td>Nr. pui morti</td>
567        <td>Gr. pui morti</td>
568        <td>Ferma</td>
569        <td>Documente</td>
570        <td>Creat</td>
571        <td>Comenzi</td>
572      </tr>
573    <?php $index=1;
574    do { ?>
575    <tr class="<?php if ($index%2) echo "lista_impar"; else echo "lista_par";
576      <td align="center"><?php echo $index; ?> </td>
577        <td><?php echo $row['Numar'];?> </td>
578        <td><?php echo date("d/m/Y", strtotime($row['Data']));?> </td>
579        <td><?php echo  $row['Pui'];?> </td>
580        <td><?php echo  $row['Greutate'];?> </td>
581        <td><?php echo  $row['Pui_Morti'];?> </td>
582        <td><?php echo  $row['Greutate_Morti'];?> </td>
583        <td><?php echo  $row['Ferma'];?> </td>
584        <td><?php echo  $row['Documente'];?> </td>
```

```
<tr class="<?php if ($index%2) echo "lista_impar"; else echo "lista_p
  <td align="center"><?php echo $index; ?> </td>
      <td><?php echo $row['Numar'];?> </td>
      <td><?php echo date("d/m/Y", strtotime($row['Data']));?> <
      <td><?php echo  $row['Pui'];?> </td>
      <td><?php echo  $row['Greutate'];?> </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 – 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 Pearl
- 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
- conversiile sunt "**umane**" **nu** numerice

```
<?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++
- Opertori
  - 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

```
$i = 1;
while ($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;
}
```

- Similare cu echivalentele C/C++

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 caracter 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[0] 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 caracter 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 “ ”

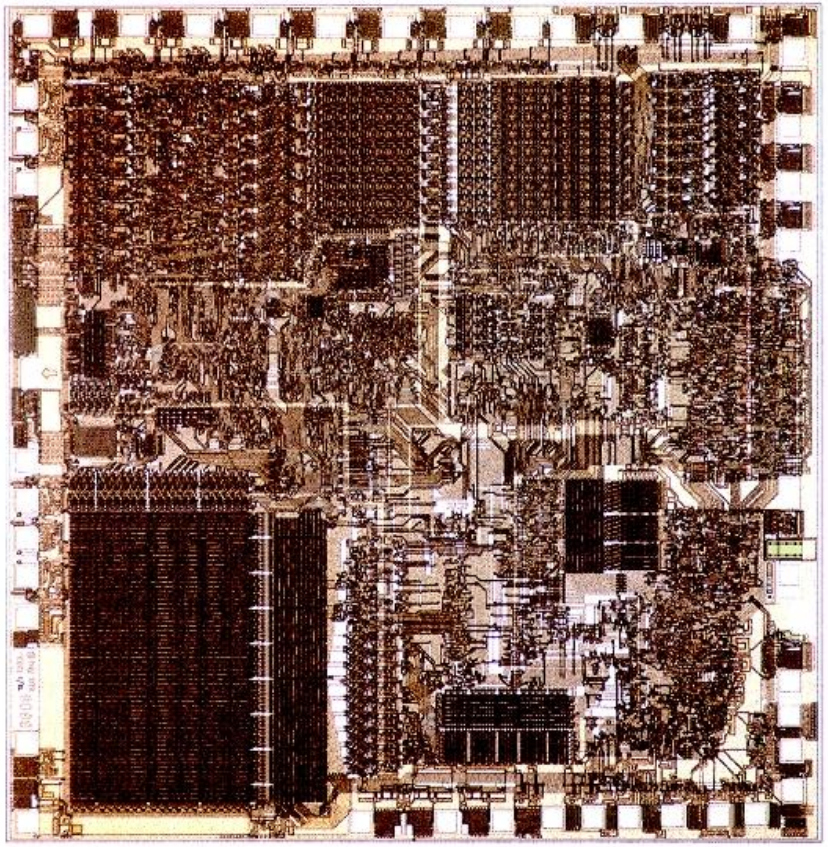
- sintaxa **complexa** pentru interpretarea variabilelor in interiorul sirurilor **{ }**

```
<?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.
?>
```

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

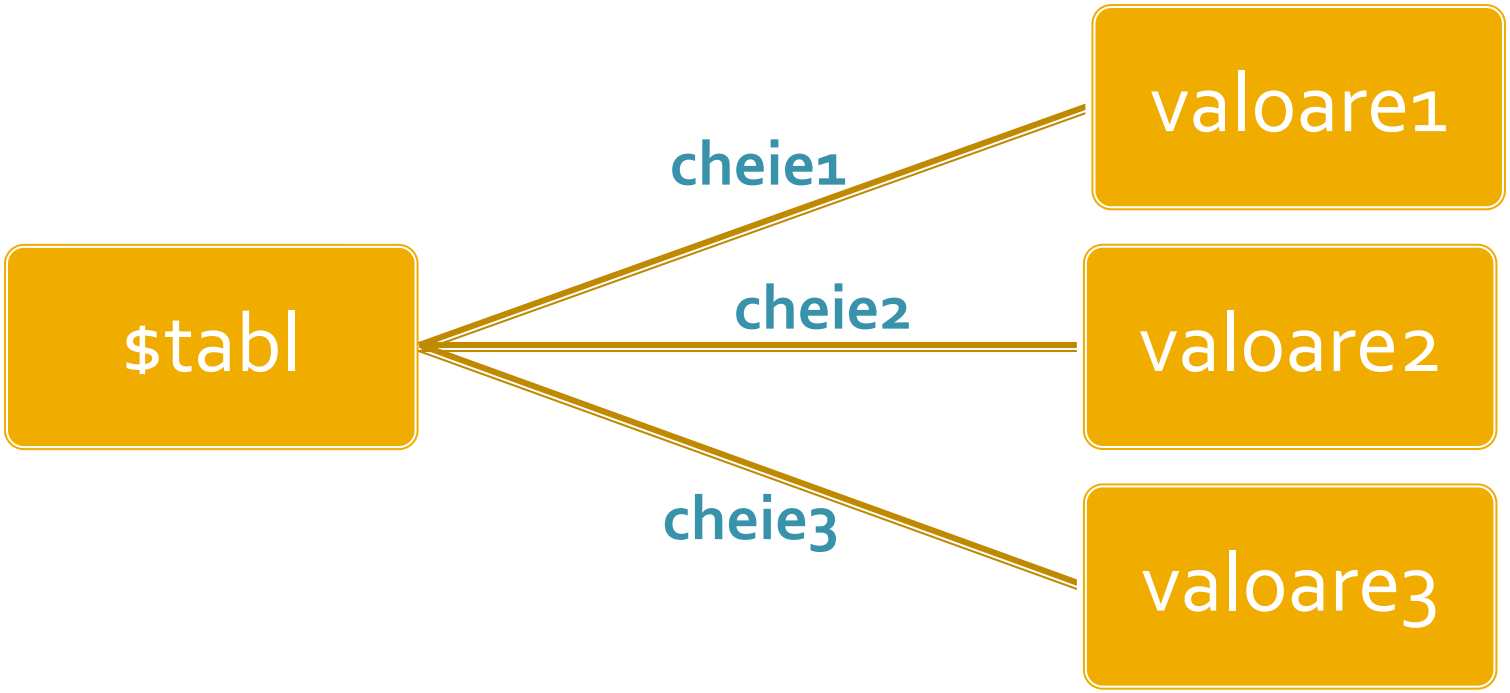
- Efectuare foarte rapida a unui numar **mic** de instructiuni, de **complexitate redusa**, repetate de un numar foarte mare de ori
- Programare: coborarea rationamentului la nivelul de **complexitate redusa**, cu obtinerea performantei prin structuri repetitive simple efectuate rapid.
- Operatii repetitive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

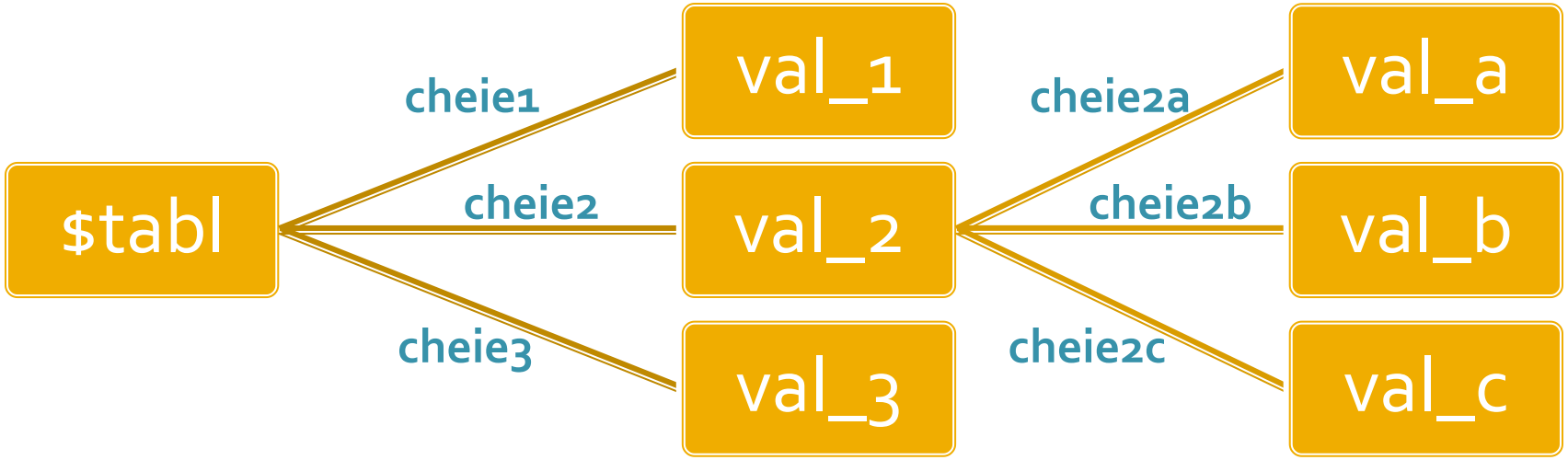
Tablou = Arbore

- $tabl = array(cheie1 => valoare1, cheie2 => valoare2, cheie3 => valoare3)



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)



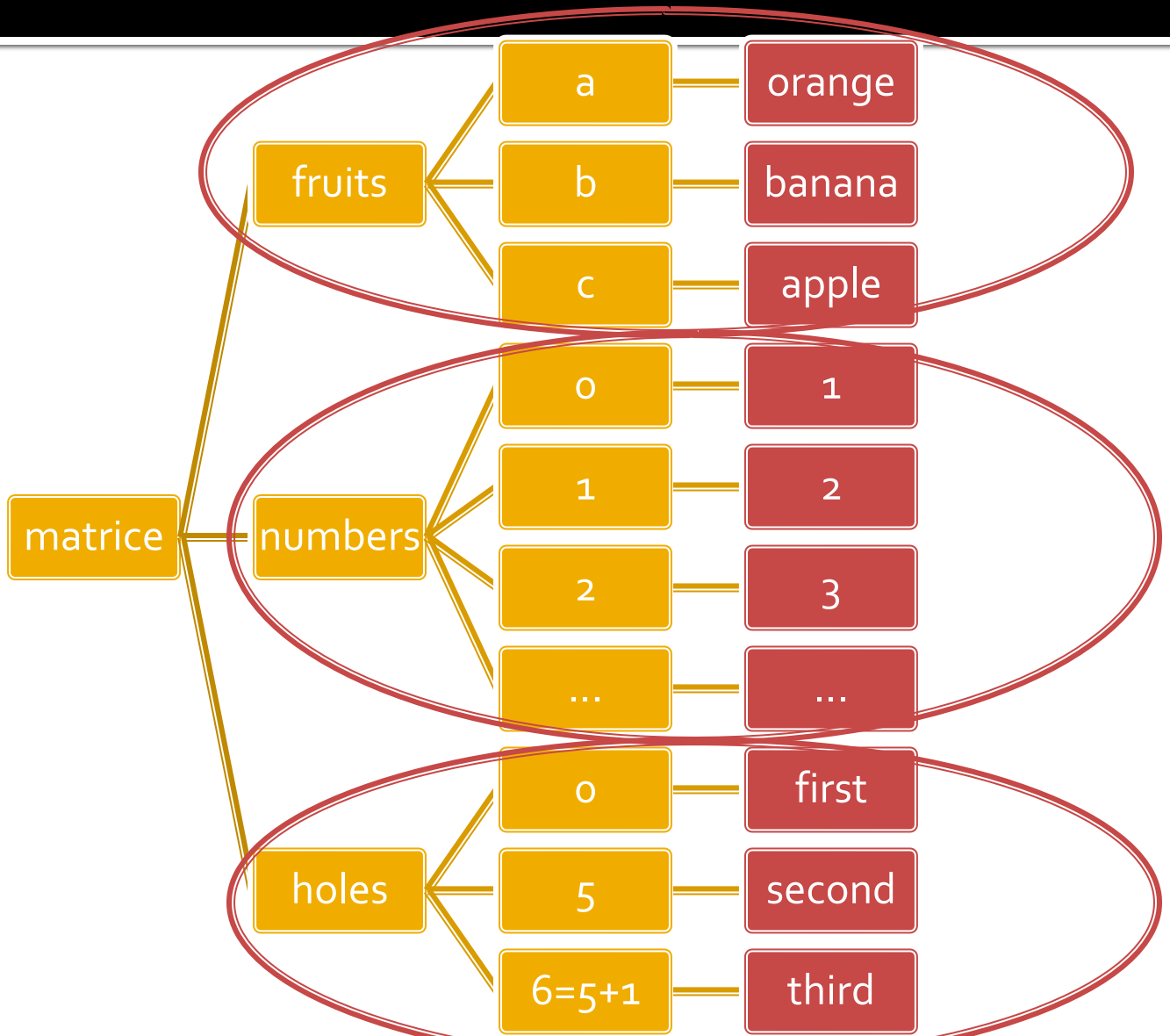
Tablouri in PHP

```
$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 daca la definirea tablourilor sunt folosite cheile numerice implicite (numere intregi)
- do … while si while se pot folosi impreuna cu functii 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 $cheie => $continut)
            echo "matr[".$cheie."]=".$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

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

Subdivizare $_POST

- atributul **name** in forma devine **cheie** in tabloul global $_POST
  - <input type="text" name="carti_cant" size="3" maxlength="3" />
  - $_POST['carti_cant'] contine valoarea introdusa
- realizand atributul **name** ca tablou, se obtine in $_POST un "subtablou" (ramificare locala a arborelui) care grupeaza elementele input
  - <input type="text" name="cant[carti]" size="3" maxlength="3" />
  - $_POST ['cant'] ['carti'] contine valoarea introdusa

Subdivizare $_POST

- realizand atributul name ca tablou, se obtine in $_POST un "subtablou" (ramificare locala a arborelui) care grupeaza elementele dorite
  - <input type="text" name="cant[carti]" size="3" maxlength="3" />
  - $_POST ['cant'] ['carti'] contine valoarea introdusa
- Este necesar pentru a grupa elementele similare pe care sa le prelucram la receptie cu foreach
- $_POST contine si alte elemente pe care le dorim eventual tratate separat
  - numele (name) si valoarea butonului "submit" apar in $_POST de exemplu

Exemplu utilizare tablouri

Exemplu

- In exemplul anterior utilizarea tablourilor va aduce urmatoarele avantaje:
  - codul va fi mai concis
  - codul va fi mai general (valabil si pentru 5 produse si pentru 1000)
  - scalabilitate crescuta (se pot adauga usor produse)

Exemplu

- fiecare produs e caracterizat de:
  - nume
  - pret
  - (eventual) descriere
  - cantitate comandata
- putem folosi unul din atribute ca si cheie (numele in exemplu)
- se poate controla (prin atributul name =“”) structura variabilei globale $_POST

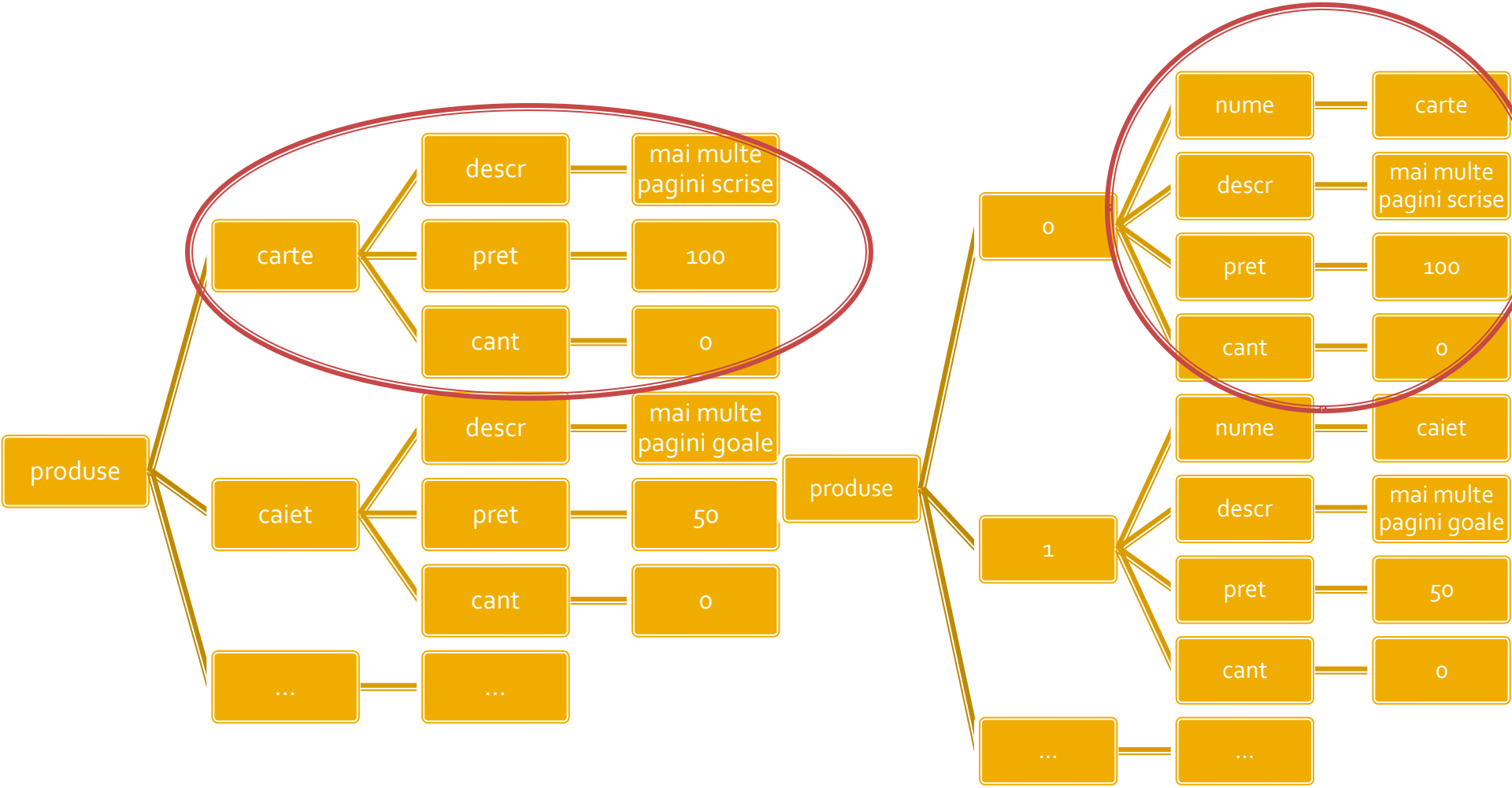
Tablou produse

- una din structurile posibile

```
$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)
                    );
```

- se urmareste obtinerea unei structuri clare
  - usor de modificat/adaugat date
  - usor de utilizat
- daca definitia se introduce in fisierul antet va fi accesibila in toate fisierele individuale

Tablou produse



Laborator – L3 – sursa 1

```
<?php
define('PRET_CARTE',100);
define('PRET_CAIET',50);
define('PRET_PENAR',150);
define('PRET_STILOU',125);
define('PRET_CREION',25);
?><h1>Magazin online Firma X SRL</h1>
<h2>Realizati comanda</h2>
<form action="rezultat.php" method="post">
<table border="0">
<tr bgcolor="#cccccc"><td>Nr.</td><td width="150">Produs</td><td width="50">Pret</td><td
width="15">Cantitate</td></tr>
<tr><td>1</td><td>Carti</td><td align="center"><?php echo PRET_CARTE;?></td><td align="center"><input
name="carte_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>2</td><td>Caiete</td><td align="center"><?php echo PRET_CAIET;?></td><td align="center"><input
name="caiet_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>3</td><td>Penare</td><td align="center"><?php echo PRET_PENAR;?></td><td align="center"><input
name="penar_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>4</td><td>Stilouri</td><td align="center"><?php echo PRET_STILOU;?></td><td align="center"><input
name="stilou_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>5</td><td>Creioane</td><td align="center"><?php echo PRET_CREION;?></td><td align="center"><input
name="creion_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr>
 <td colspan="4" align="center">
<input type="submit" value="Trimite" /></td></tr>
</table>
</form>
```

Crearea listei de produse

```
<?php require('antet.php');?>
<h2>Lista Produse</h2>
<table border="1">
<tr bgcolor="#cccccc"><td>Nr.</td><td width="150">Produs</td><td width="150">Descriere</td><td
width="50">Pret</td></tr>
<?php
$index=1;
foreach ($produse as $prod => $detalii)
                {
                echo "<tr><td>".$index."</td><td>".ucfirst(strtolower($prod))."</td><td>".$detalii['descr']."</td><td
align=\"center\">".$detalii['pret']."</td></tr>";
                $index++;
                }
?>
<?php
$index=1;
foreach ($produse as $prod => $detalii)
                {?>
<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++;
                } ?>
<tr><td colspan="4" align="center"><a href="formular.php">Comanda</a></td></tr></table>
<?php require('subsol.php');?>
```

Crearea listei de produse

Magazin Firma X SRL

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Descriere	Pret
1	Carte	mai multe pagini scrise legate	100
2	Caiet	mai multe pagini goale legate	50
3	Penar	loc de depozitat instrumente de scris	150
4	Stilou	instrument de scris albastru	125
5	Creion	instrument de scris gri	25
1	Carte	mai multe pagini scrise legate	100
2	Caiet	mai multe pagini goale legate	50
3	Penar	loc de depozitat instrumente de scris	150
4	Stilou	instrument de scris albastru	125
5	Creion	instrument de scris gri	25
Comanda			

Subdivizare $_POST

- atributul **name** in forma devine **cheie** in tabloul global $_POST
  - <input type="text" name="carti_cant" size="3" maxlength="3" />
  - $_POST['carti_cant'] contine valoarea introdusa
- realizand atributul **name** ca tablou, se obtine in $_POST un "subtablou" (ramificare locala a arborelui) care grupeaza elementele input
  - <input type="text" name="cant[carti]" size="3" maxlength="3" />
  - $_POST ['cant'] ['carti'] contine valoarea introdusa

Subdivizare $_POST

- realizand atributul name ca tablou, se obtine in $_POST un "subtablou" (ramificare locala a arborelui) care grupeaza elementele dorite
  - <input type="text" name="cant[carti]" size="3" maxlength="3" />
  - $_POST ['cant'] ['carti'] contine valoarea introdusa
- Este necesar pentru a grupa elementele similare pe care sa le prelucram la receptie cu foreach
- $_POST contine si alte elemente pe care le dorim eventual tratate separat
  - numele (name) si valoarea butonului "submit" apar in $_POST de exemplu

Crearea formei de comanda

```
<?php require('antet.php');?>
<h2>Realizati comanda</h2>
<form action="rezultat.php" method="post">
<table border="0">
<tr bgcolor="#cccccc"><td>Nr.</td><td width="150">Produs</td><td width="50">Pret</td><td
width="15">Cantitate</td></tr>
<?php
$index=1;
foreach ($produse as $prod => $detalii)
                {?>
<tr><td><?php echo $index;?></td><td><?php echo ucfirst(strtolower($prod));?></td><td
align="center"><?php echo $detalii['pret'];?></td><td><input name="<?php echo
"cant[".$prod."]";?>" type="text" value="0" size="3" maxlength="3" /></td></tr>
<?php $index++;
                } ?>
<tr><td colspan="4" align="center"><input type="submit" value="Trimite" /></td></tr>
</table>
</form>
<?php require('subsol.php');?>
```

Crearea listei de produse

Magazin Firma X SRL

Magazin online Firma X SRL

Realizati comanda

Nr.	Produs	Pret	Cantitate
1	Carte	100	0
2	Caiet	50	0
3	Penar	150	0
4	Stilou	125	0
5	Creion	25	0

Trimite

Prelucrarea comenzii

```
<?php require('antet.php');?>
<h2>Rezultate comanda</h2>
<p>Pret total (fara TVA):
<?php
$pret=0;
$afis="";
foreach ($_POST['cant'] as $prod => $cant)
                {
                $pret += $cant*$produse[$prod]['pret'];
                $afis .= "+".$cant."x".$produse[$prod]['pret'];
                }
echo $pret;
?>
<p>Obtinut astfel: <?php echo $afis;?></p>
<p>Pret total (cu TVA): <?php echo $pret*1.19;?></p>
<p><?php
echo "<pre>";
print_r ($_POST);
echo "</pre>";
?>
</p>
<p>Comanda receptionata la data: <?php echo date('d/m/Y')." ora ".date('H:i');?></p>
<?php require('subsol.php');?>
```

Prelucrarea comenzii

Magazin Firma X SRL

Magazin online Firma X SRL

Rezultate comanda

Pret total (fara TVA): 600

Obtinut astfel: +2x100+2x50+2x150+0x125+0x25

Pret total (cu TVA): 714

```
Array
(
    [cant] => Array
        (
            [carte] => 2
            [caiet] => 2
            [penar] => 2
            [stilou] => 0
            [creion] => 0
        )

)
```

Comanda receptionata la data: 17/03/2010 ora 13:55

Acces la server-ul MySql din PHP

Acces la server-ul MySql din PHP

- Bibliotecile corespunzatoare trebuie activate in php.ini – vezi laboratorul 1.
  - mysql
  - mysqli (improved accesul la functionalitati ulterioare MySql 4.1)
- O baza de date existenta poate fi accesata daca exista un utilizator cunoscut in PHP cu drepturi de acces corespunzatoare – vezi laboratorul 1.
- O baza de date poate fi creata si din PHP dar nu e metoda recomandata daca nu e necesara
  - cod dificil de implementat pentru **o singura** utilizare
  - necesita existenta unui utilizatori cu drepturi mai mari pentru crearea bazei de date si alocarea de drepturi unui utilizator restrans

Functii PHP de acces MySql

- mysql_query
  - trimiterea unei interogari SQL spre server
  - resource **mysql_query** (string query [, resource link_identifier])
  - rezultatul
    - SELECT, SHOW, DESCRIBE sau EXPLAIN – resursa (tabel)
    - UPDATE, DELETE, DROP, etc – true/false
- mysql_fetch_assoc
  - returneaza o **matrice asociativa** corespunzatoare liniei de la indexul intern (indecsi de tip sir corespunzatori denumirii coloanelor – field – din tabelul de date) si incrementeaza indexul intern sau **false** daca nu mai sunt linii
  - array **mysql_fetch_assoc** (resource result)

Functii PHP de acces MySql Parcurgerea resurselor rezultat

- mysql_fetch_assoc
  - returneaza o **matrice asociativa** corespunzatoare liniei de la indexul intern (indecsi de tip sir corespunzatori denumirii coloanelor – field – din tabelul de date) si incrementeaza indexul intern sau **false** daca nu mai sunt linii
  - array **mysql_fetch_assoc** (resource result)
- mysql_fetch_row
  - returneaza o matrice cu indecsi intregi
  - array mysql_fetch_row (resource result)

Functii PHP de acces MySql Parcurgerea resurselor rezultat

- mysql_fetch_array
  - grupeaza functionalitatea mysql_fetch_assoc si mysql_fetch_row
  - array **mysql_fetch_array** (resource result [, int result_type])
  - MYSQL_ASSOC, MYSQL_NUM, MYSQL_BOTH (implicit)
- mysql_data_seek
  - muta indexul intern la valoarea indicata
  - bool **mysql_data_seek** (resource result, int row_number)

Resurse MySQL

- Resursele reprezinta o combinatie intre
  - date structurate (valori + structura) rezultate in urma unor interogari SQL
  - functii de acces la aceste date/structuri
- Analogie cu POO
  - o "clasa speciala" creata in urma interogarii cu functii predefinite de acces la datele respective

Resurse MySQL

Structura

Index intern	Col 1 (tip date)	Col 2 (tip date)	....
1			
2			
...			

Date

Index intern	Col 1	Col 2	....
1	Val 11	Val 12	...
2	Val 21	Val 22	...
...	...	...	...

Functii de acces la structura

Functii de acces la date

~~Acces direct~~

Resurse MySQL

- Functiile de acces la structura sunt rareori utilizate
  - majoritatea aplicatiilor sunt concepute pe structura fixa, si cunosc structura datelor primite
  - exceptie: aplicatii generale, ex.: PhpMyAdmin
- Majoritatea functiilor de acces la date sunt caracterizate de acces secvential
  - se citesc in intregime valorile stocate pe o linie
  - simultan se avanseaza indexul intern pe urmatoarea pozitie, pregatindu-se urmatoarea citire

Resurse MySQL

- Functiile sunt optimizate pentru utilizarea lor intr-o structura de control **do { } while()**, sau **while() { }** de control
  - returneaza FALSE cand "s-a ajuns la capat"
- tipic se realizeaza o citire (mysql_fetch_assoc) urmata de o bucla **do { } while()**
  - pentru a se putea introduce cod de detectie probleme rulat o singura data

Exemplu de utilizare

```
$hostname = "localhost";
$database = "world";
$username = "web";
$password = “ceva";
$conex= mysql_connect($hostname, $username, $password);
mysql_select_db($database, $ conex);
```

```
$query = "SELECT `Code`,`Name`,`Population` FROM `country` AS c ";
$result = mysql_query($ query, $ conex) or die(mysql_error());
$row_result = mysql_fetch_assoc($ result );
$totalRows_ result = mysql_num_rows($ result );
```

Exemplu de utilizare

```
<?php
do { ?>
<tr>
        <td><?php echo $index; ?> </td>
        <td><?php echo $ row_result ['Code']; ?> </td>
        <td><?php echo $ row_result ['Name']; ?> </td>
        <td><?php echo $ row_result ['Population']; ?> </td>
</tr>
<?php
        $index++;
        }
while ($ row_result = mysql_fetch_assoc($ result )); ?>
```

Modificari laborator cu date stocate text

- Codul aplicatiei ramane in mare parte acelasi
- Se modifica doar citirea valorilor pentru popularea matricii $produse ("antet.php")

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

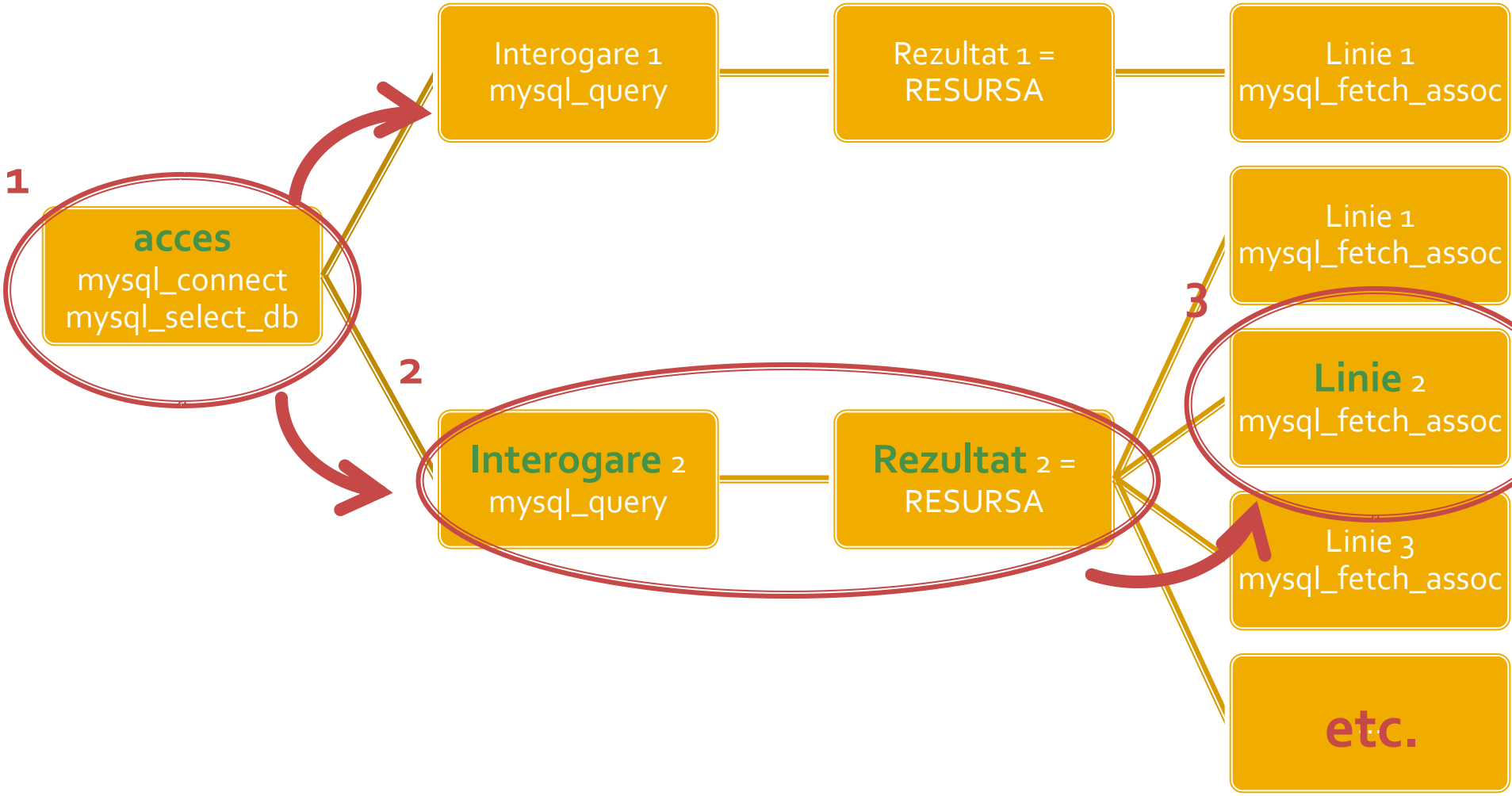
Modificari laborator cu date stocate XML

```
$xml = simplexml_load_file("lista.xml");
if ($xml)
{
foreach ($xml->categorie as $categorie)
        {
        $produse[(string)$categorie["nume"]]=array();
        foreach ($categorie->produs as $prod_cur)
                {
                $produse[(string)$categorie["nume"]][(string)$prod_cur->nume]=array
                ("descr" => (string)$prod_cur->desc, "pret" => (string)$prod_cur->pret,
                "cant" => (string)$prod_cur->cant);
                }
        }
}
```

Modificari laborator cu date stocate MySQL

```
$hostname = "localhost";
$database = "tmpaw";
$username = "web";
$password = "test";
$conex= mysql_connect($hostname, $username, $password);
mysql_select_db($database, $conex);
$query = "SELECT * FROM `categorii` AS c";
$result_c = mysql_query($query, $conex) or die(mysql_error());
$row_result_c = mysql_fetch_assoc($result_c);
$totalRows_result = mysql_num_rows($result_c);
do {
        $query = "SELECT * FROM `produse` AS p WHERE `id_categ` = ".$row_result_c['id_categ'];
        $result_p = mysql_query($query, $conex) or die(mysql_error());
        $row_result_p = mysql_fetch_assoc($result_p);
        $totalRows_result = mysql_num_rows($result_p);
        $produse[$row_result_c['nume']]=array();
        do {
                $produse[$row_result_c['nume']][$row_result_p['nume']]=array ("descr" =>
$row_result_p['detalii'], "pret" => $row_result_p['pret'], "cant" => $row_result_p['cant']);
                }
        while ($row_result_p = mysql_fetch_assoc($result_p));
        }
while ($row_result_c = mysql_fetch_assoc($result_c));
```

Functii de acces la server-ul MySql



!! IMPORTANT

PHP > 5.5

PHP 5.5

- Incapand cu versiunea 5.5 a PHP extensia mysql este declarata **depreciata**
  - orice utilizare a unei functii genereaza eroare de tip **E_DEPRECATED**
  - se preconizeaza ca in PHP > 6 aceasta extensie va fi eliminata total (**realizat**)
- Alternativele de utilizare sunt
  - extensia mysqli (MySQL Improved)
  - extensia PDO (PHP Data Objects)

Extensia mysqli

- Inafara securitatii sporite ofera acces la facilitatile curente ale server-ului MySQL
  - accesul la interogari predefinite (Prepared Statements) (viteza, securitate)
    - server side
    - client side
  - proceduri stocate pe server (viteza, securitate)
  - interogari multiple
  - tranzactii (integritate)

Extensia mysqli

- Doua modalitati de utilizare
  - procedurala (similar mysql)
  - POO (similar PDO)
- Utilizarea procedurala (aproape) similara cu utilizarea extensiei originale mysql
  - tranzitie facila
  - tranzitie cu mici diferente de parametri

mysqli – Procedural

```
<?php
$mysqli = mysqli_connect("example.com", "user", "password", "database");
$res = mysqli_query($mysqli, "SELECT 'Please do not use the mysql extension ' AS _msg FROM DUAL");
$row = mysqli_fetch_assoc($res);
echo $row['_msg'];

$mysql = mysql_connect("example.com", "user", "password");
mysql_select_db("test");
$res = mysql_query("SELECT ' for new developments.' AS _msg FROM DUAL", $mysql);
$row = mysql_fetch_assoc($res);
echo $row['_msg'];
?>
```

- toate functiile mysql au un echivalent mysqli
- majoritatea functiilor au aceeasi parametri in aceeasi ordine
- sunt totusi functii cu mici diferente (Ex: **mysqli_connect**, **mysqli_query**)

mysqli – Programare orientata obiect

```
<?php
$var = new mysqli("example.com", "user", "password", "database");
$res = $var->query ($mysqli, "SELECT 'Please do not use the mysql extension ' AS _msg FROM DUAL");
$row = $res->fetch_assoc();
echo $row['_msg'];

$mysql = mysql_connect("example.com", "user", "password");
mysql_select_db("test");
$res = mysql_query("SELECT ' for new developments.' AS _msg FROM DUAL", $mysql);
$row = mysql_fetch_assoc($res);
echo $row['_msg'];
?>
```

Resurse MySQL – mysqli

Structura

Index intern	Col 1 (tip date)	Col 2 (tip date)	....
1			
2			
...			

Date

Index intern	Col 1	Col 2	....
1	Val 11	Val 12	...
2	Val 21	Val 22	...
...	...	...	...

Metode

Constructor	query	fetch_assoc	....

Functii de acces la structura

Functii de acces la date

~~Acces direct~~

Metode atasate resursei

Conversia la mysqli (obligatorie)

- exemplul anterior

```
$hostname = "localhost";
$database = "tmpaw";
$username = "web";
$password = "test";
$conex= mysql_connect($hostname, $username, $password);
mysql_select_db($database, $conex);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
                LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
$result = mysql_query($query, $conex) or die(mysql_error());
$row_result = mysql_fetch_assoc($result);
$totalRows_result = mysql_num_rows($result);

do {
                $produse[$row_result['nume_categ']][$row_result['nume']]=array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
                }
while ($row_result = mysql_fetch_assoc($result));
```

mysqli (Procedural)

```
//$conex= mysql_connect($hostname, $username, $password);
//mysql_select_db($database, $conex);
$conex = mysqli_connect($hostname, $username, $password, $database);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
                LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
//$result = mysql_query($query, $conex) or die(mysql_error());
$result = mysqli_query($conex, $query);

//$row_result = mysql_fetch_assoc($result);
$row_result = mysqli_fetch_assoc($result);

//$totalRows_result = mysql_num_rows($result);
$totalRows_result = mysqli_num_rows($result);

do {
                $produse[$row_result['nume_categ']][$row_result['nume']]=array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
                }
//while ($row_result = mysql_fetch_assoc($result));
while ($row_result = mysqli_fetch_assoc($result));
```

mysqli (POO)

```
//$conex= mysql_connect($hostname, $username, $password);
//mysql_select_db($database, $conex);
//$conex = mysqli_connect($hostname, $username, $password, $database);
$conex = new mysqli($hostname, $username, $password, $database);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
                LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
//$result = mysql_query($query, $conex) or die(mysql_error());
//$result = mysqli_query($conex, $query);
$result = $conex->query( $query );

//$row_result = mysql_fetch_assoc($result);
//$row_result = mysqli_fetch_assoc($result);
$row_result = $result->fetch_assoc();

//$totalRows_result = mysql_num_rows($result);
//$totalRows_result = mysqli_num_rows($result);
$totalRows_result = $result->num_rows;

do {
                $produse[$row_result['nume_categ']][$row_result['nume']]=array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
                }
//while ($row_result = mysql_fetch_assoc($result));
while ($row_result = $result->fetch_assoc());
```

Contact

- Laboratorul de microunde si optoelectronica
- http://rf-opto.etti.tuiasi.ro
- rdamian@etti.tuiasi.ro