

Curs 4

Tehnici moderne de proiectare a aplicatiilor web

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

LABORATOR

I.	Implementarea unui sistem de dezvoltare a aplicatiilor Web, instalare PHP, MySql, Apache si legaturile dintre ele	2 ore
II	Design web avansat folosind CSS	2 ore
III	Interogarea unei baze de date. Exercitii SQL	2 ore
IV	Utilizare PHP I	2 ore
V	Utilizare PHP II	2 ore
VI	Utilizare PHP pentru accesul la o baza de date	2 ore
VII	Aplicatie distribuita complexa	2 ore
	Total	14 ore

Nota

- An V
 - 33% E
 - 66% Aplicatii
 - 33% L
 - 33% P

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

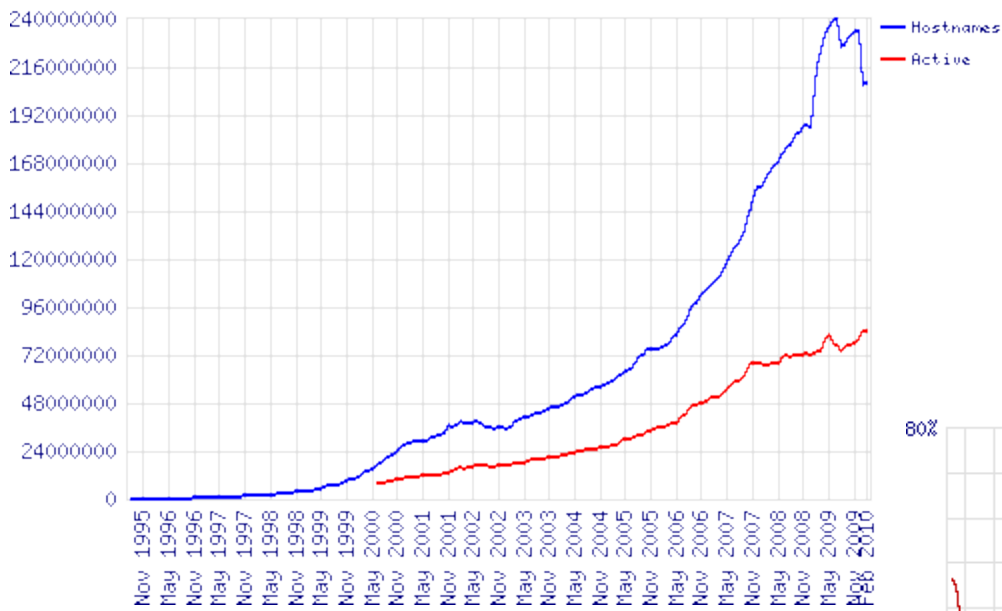
PHP

- Hypertext PreProcessor - acronim recursiv
 - initial – Personal Home Page / Form Interpreter
 - 1995 – 1.0
- versiune curenta: 5.3.2
 - 2010-03-04
- limbaj de scripting de uz general,
- rulare pe server (server-side scripting)
- open source

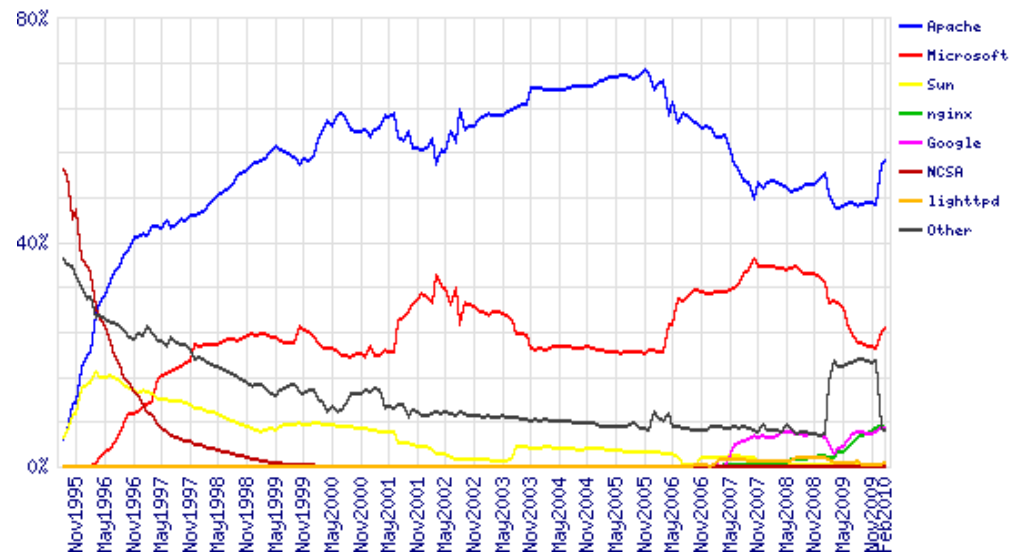
TIOBE Programming Community Index for March 2010

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

Numar de server-e



Numar de server-e existente / active



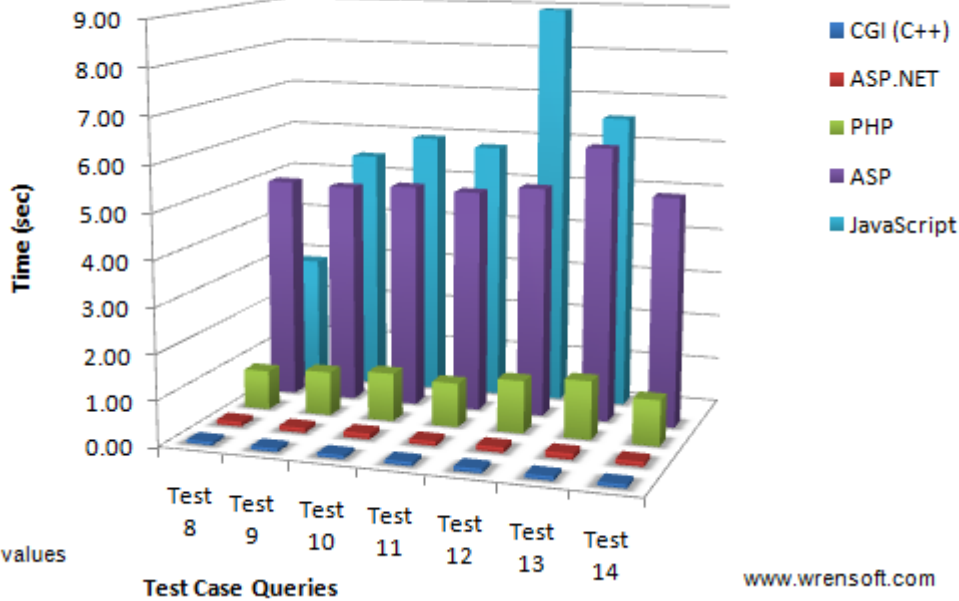
Aplicatia de server web activa [%]

Avantaje PHP

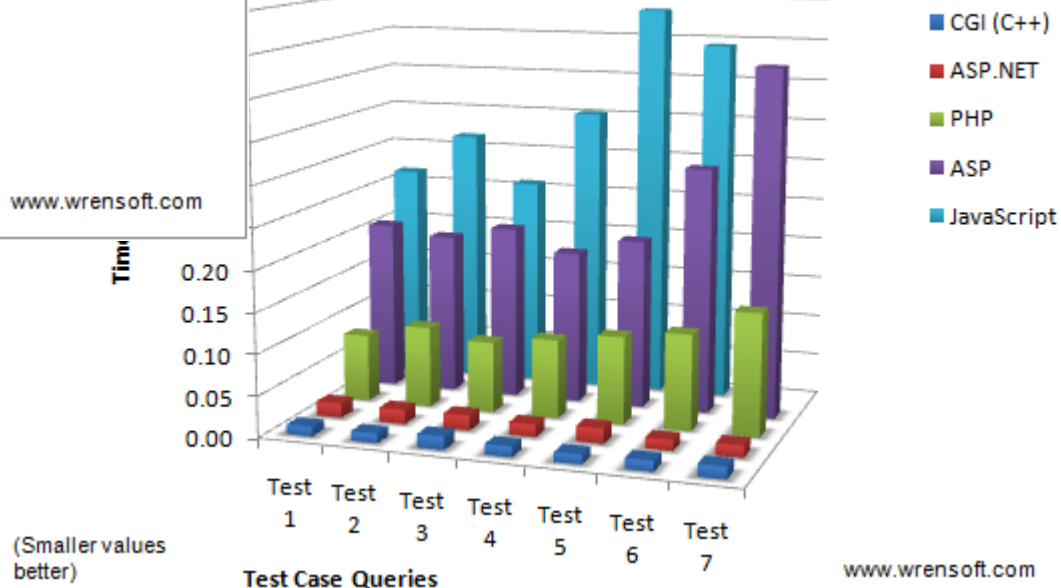
- Performanta ridicata
- Interfata cu multe sisteme de baze de date
- Costuri reduse
- Biblioteci incluse pentru majoritatea operatiunilor uzuale in aplicatii web
- Usurinta in invatare (C)
- Portabilitate
 - Disponibilitate a codului sursa
- Exemple disponibile in comunitate
- Suport disponibil

Performanta

Zoom search benchmarking
(Medium site - 60K pages, 2.5 million words)

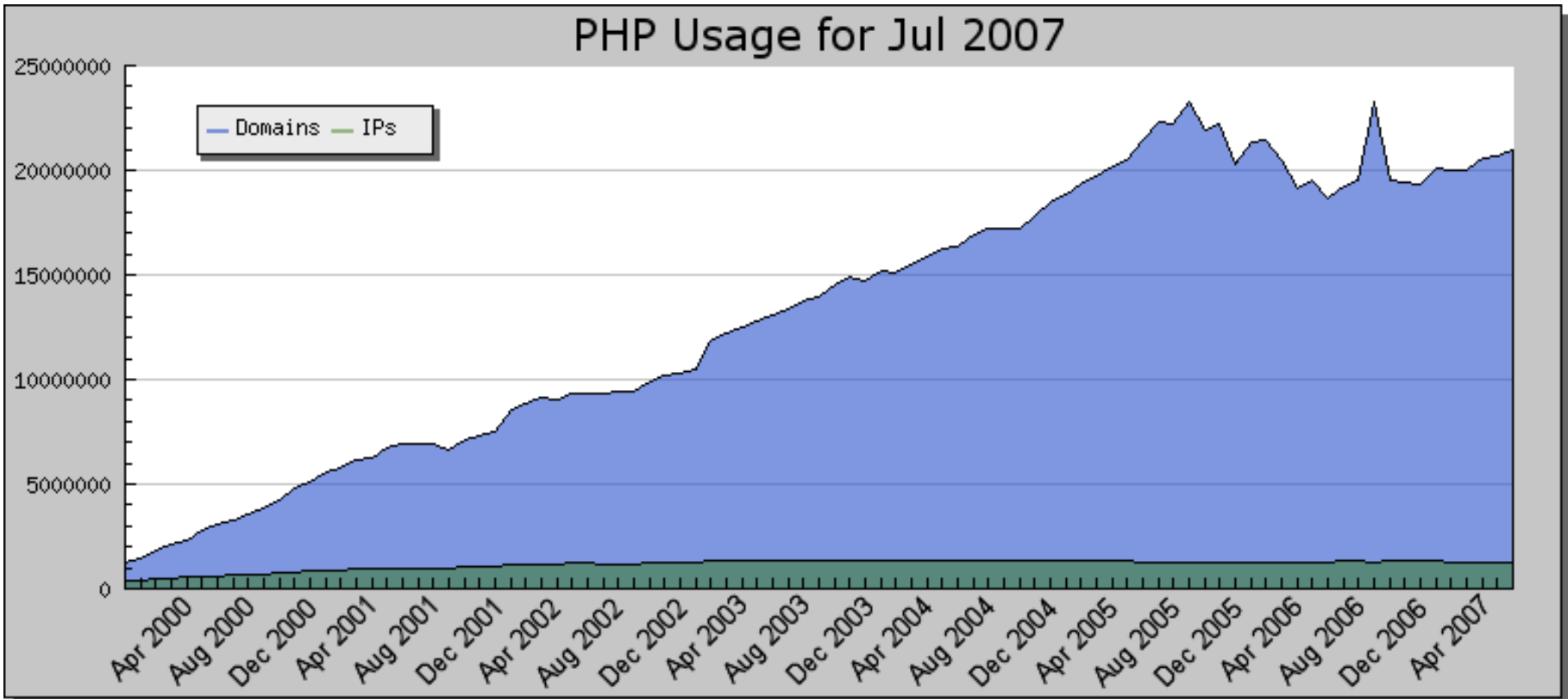


Zoom search benchmarking
(Small site - 400 pages, 266K words)

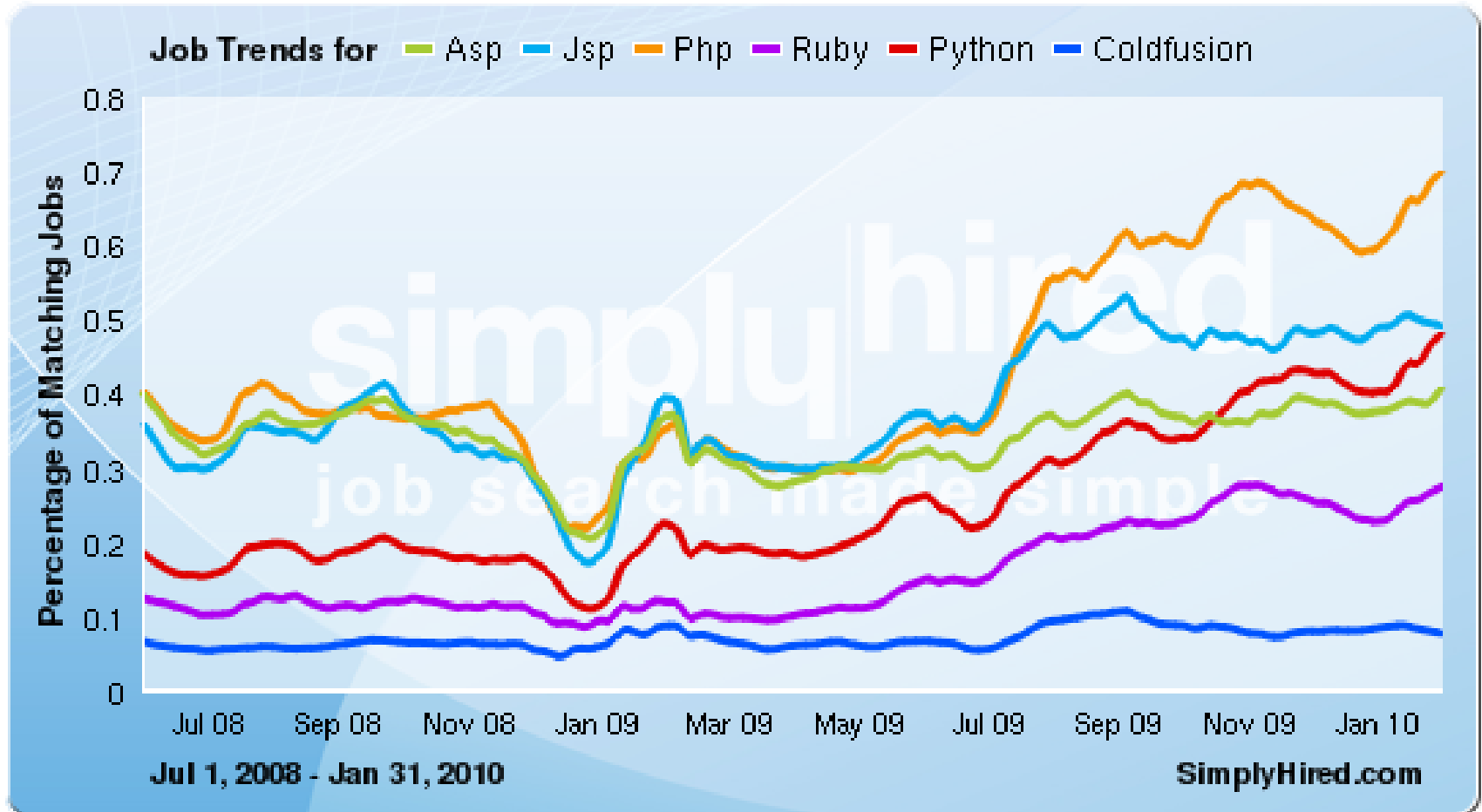


Raspiandire

PHP Usage for Jul 2007



Angajare



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
- orientat spre aplicatii web dinamice (biblioteci)
- poate fi integrat in HTML – utilizarea tipica

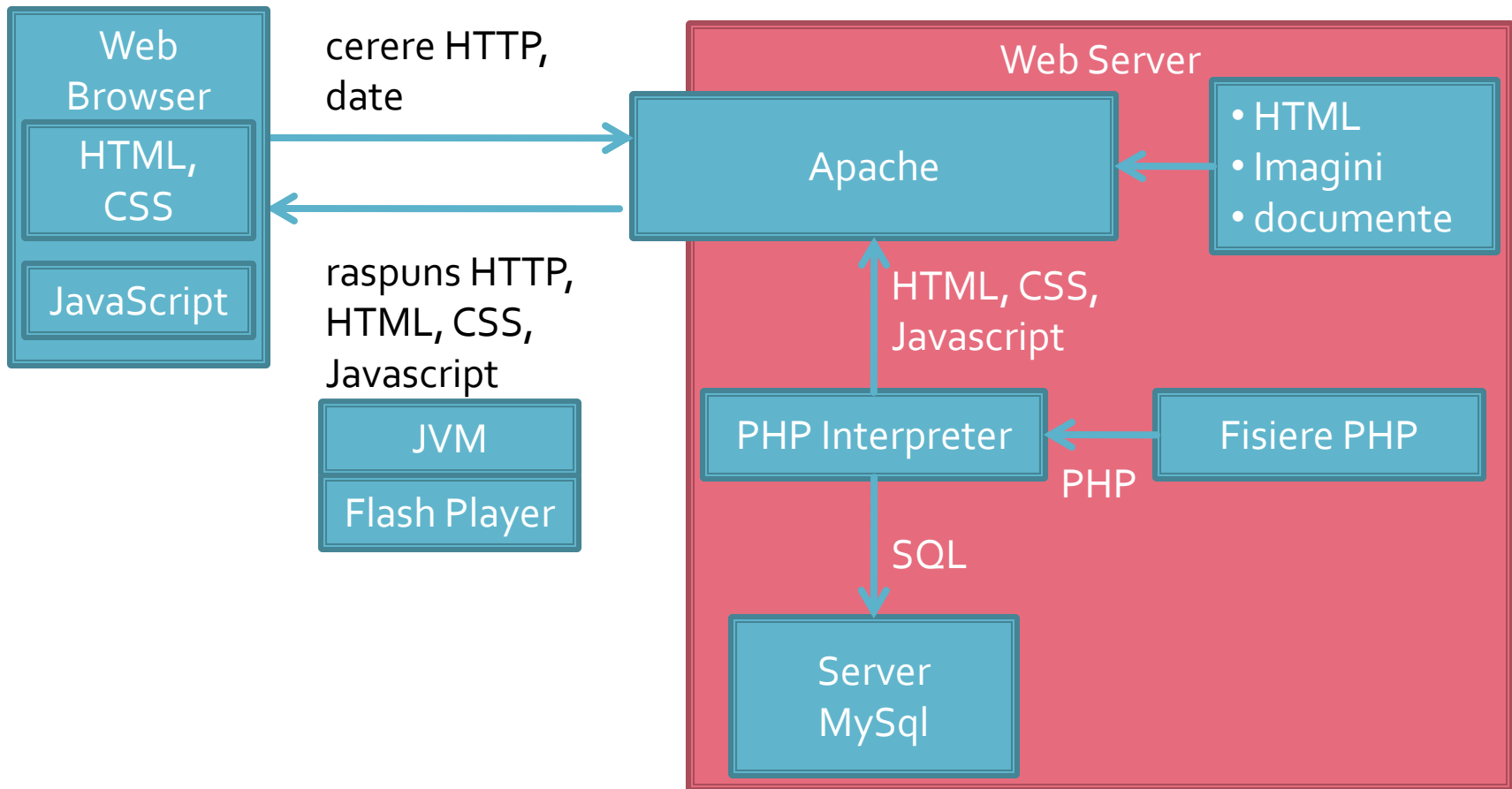
Separare cod PHP

- 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
- `<script language="php"> ... </script>`
 - stil script, disponibil
- `<% ... %>`
 - stil ASP, de obicei dezactivat

Interactiune client/server



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"/>

Trimite

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

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>
<?php echo '<p>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>
```

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 serverul web

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 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

- tipul de date nu e decis de programator prin declaratia variabilei
- e decis de interpretor in functie de tipul de date stocat in variabila respectiva
- declaratia 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

- 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

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

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

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 = 0;  
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

Elemente de control – foreach

- foreach() este elementul de control pentru a itera elementele unei **matrici**
- matricea 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.

Matrici 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[0]=nedefinit
```

```
$matr[1]=nedefinit
```

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

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

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

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

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

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

Afisarea matricilor

```
$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  
  )  
)
```

Elemente de control – foreach

- `foreach (array_expression as $key => $value) statement`
- `foreach (array_expression as $value) statement`
- iterarea prin fiecare element al matricii
- la fiecare element variabila declarata in instructiune **\$key** ofera acces la cheia curenta iar variabila **\$value** ofera acces la valoarea asociata
- `foreach()` lucreaza cu o **copie** a matricii deci matricea originala nu va fi modificata de prin schimbarea 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
```

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

Laborator

- 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 - continuare

- se creaza macar 3 pagini:
 - lista produse
 - formular comanda
 - rezultat
- forma paginilor:
 - tabel

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	

Laborator - suplimentar

- pentru usurinta modificarilor ulterioare se lucreaza cu matrici
- forma paginilor:
 - tabel controlat prin CSS

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

Contact

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