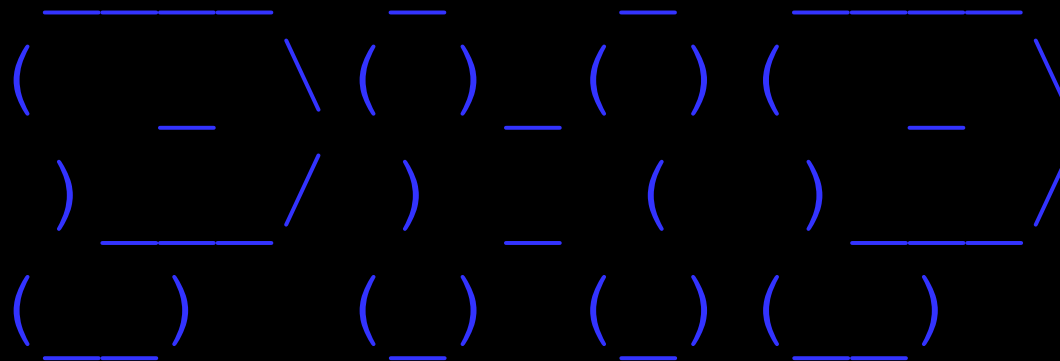


Tehnologii Web

programare Web



dezvoltarea aplicațiilor Web în PHP

„E mediocru ucenicul
care nu-și depășește maestrul.”

Leonardo da Vinci

Personal Home Page Tools (1995)

Rasmus Lerdorf

PHP 3 (1998)

dezvoltat de Zend – **Zeev Suraski & Andi Gutmans**

PHP 4 (2000)

suport pentru programare obiectuală

PHP 5 (2004) – varianta cea mai recentă: **PHP 5.6 (2014)** ▶
noi facilități inspirate de Java

PHP 6 (actualmente abandonat)

phpng ▶ PHP 7 (2015)

strong typing, suport pentru Unicode, performanță,...

php: caracterizare

Server de aplicații Web

oferă un limbaj de programare
de tip *script*, interpretat

poate fi inclus direct și în cadrul documentelor HTML

php: caracterizare

Limbajul PHP este procedural, oferind suport și pentru alte paradigme de programare (obiectuală și, mai recent, funcțională)

php: caracterizare

Limbajul PHP este procedural, oferind suport și pentru alte paradigme de programare (obiectuală și, mai recent, funcțională)

poate fi folosit și ca limbaj de uz general

php: caracterizare

Sintaxă inspirată de C, Perl și Java – *case sensitive*

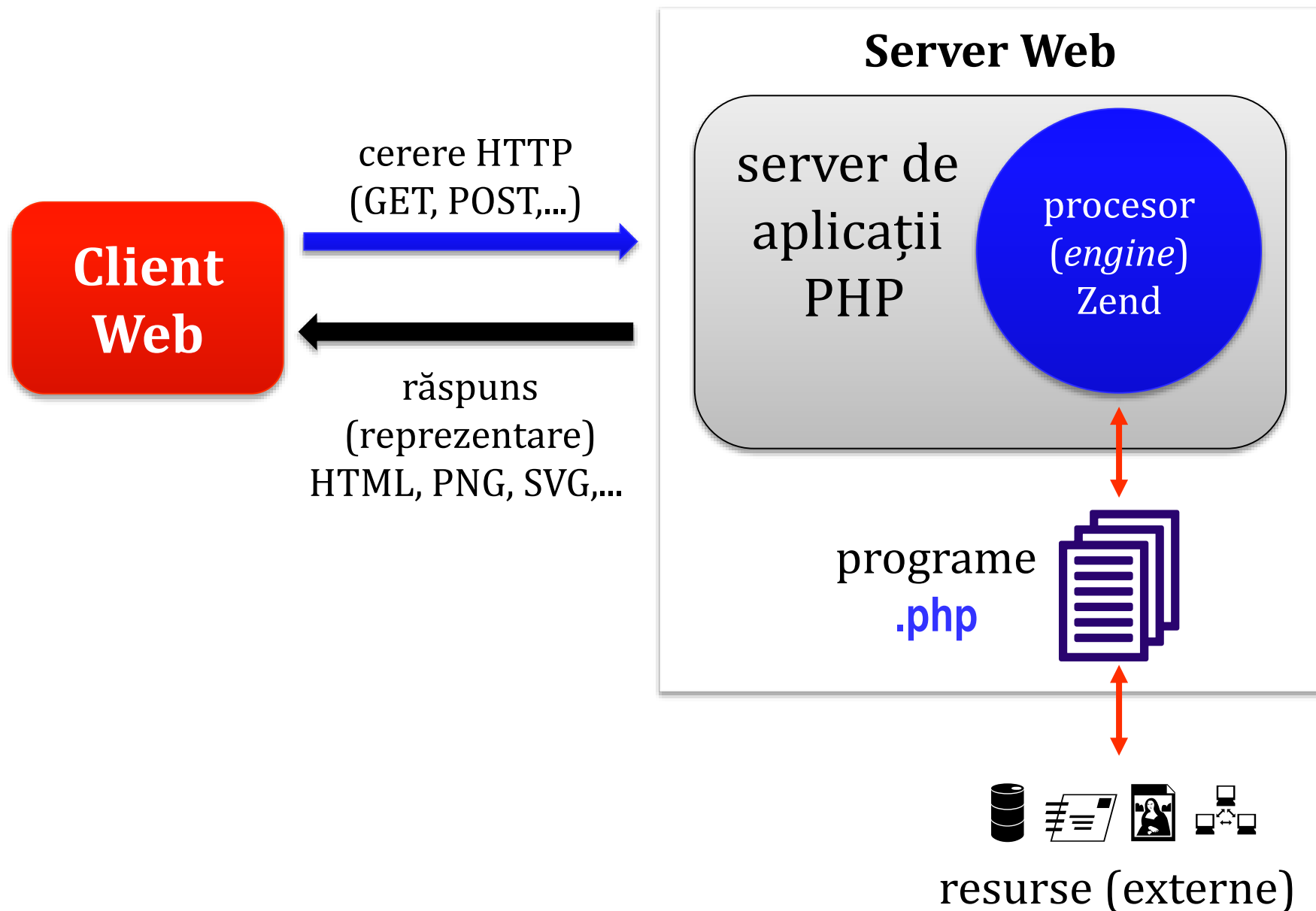
uzual, programele PHP au extensia **.php**

php: caracterizare

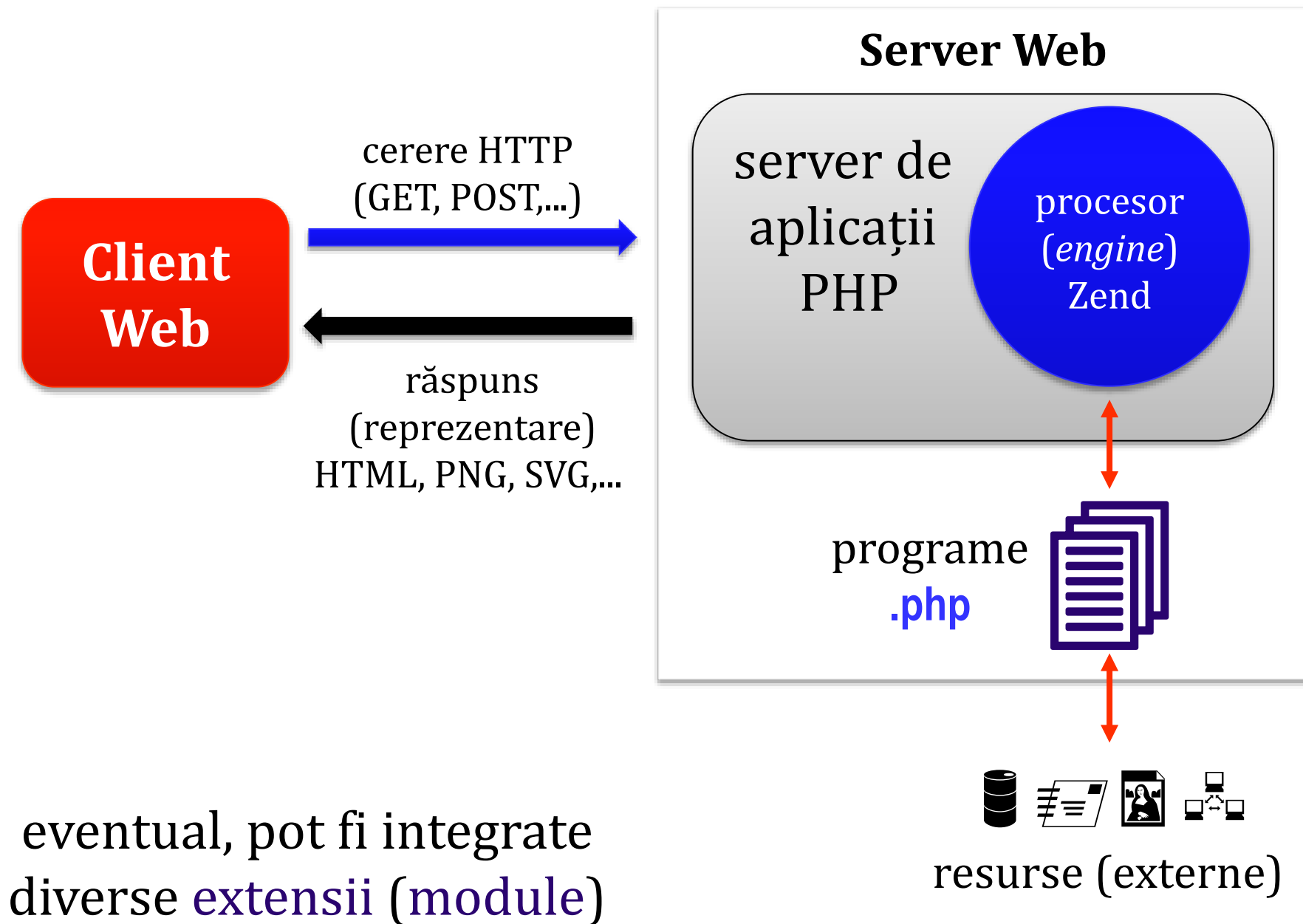
Disponibil gratuit – *open source* – pentru diverse platforme (Linux, Windows, Mac OS X, UNIX) și servere Web: Apache, IIS, nginx,...

www.php.net
www.zend.com

Maniera de funcționare a procesorului (*engine*-ului) PHP



Maniera de funcționare a procesorului (*engine*-ului) PHP



php: caracterizare

Programul PHP e interpretat de Zend Engine 2
generând instrucțiuni (*opcodes*) interne

<http://www.php.net/manual/en/internals2.opcodes.php>
<http://www.phpinternalsbook.com/>

precision = 14 ; controlul preciziei valorilor float – detalii la <http://php.net/precision>
safe_mode = Off ; controlul procesării – <http://php.net/safe-mode>
max_execution_time = 30 ; număr maxim de secunde privind execuția unui program
memory_limit = 128M ; dimensiunea maximă a memoriei alocate unui script
post_max_size = 8M ; dimensiunea maximă a datelor transmise prin metoda POST
default_mimetype = "text/html" ; tipul MIME implicit transmis de un script PHP
file_uploads = On ; sunt permise *upload*-uri de fișiere
upload_max_filesize = 32M ; dimensiunea maximă a unui fișier preluat de la client
allow_url_fopen = On ; se permite deschiderea de fișiere specificate printr-un URL
session.use_cookies = 1 ; sesiunile Web vor recurge la *cookie*-uri
session.name = PHPSESSID ; numele *cookie*-ului referitor la sesiunea Web
...
; precizarea extensiilor încărcate la inițializarea serverului de aplicații
extension=php_pdo_sqlite.dll
extension=php_mysql.dll
extension=php_soap.dll
...

diverse comportamente ale platformei PHP,
inclusiv încărcarea extensiilor (biblioteci partajate .so/.dll),
se pot configura via fișierul [php.ini](#)

php: caracterizare

Maniera de rulare a programelor PHP
– eventual, la nivel de bloc de cod –
poate fi ajustată via directiva **declare**

// setul de caractere folosit pentru generarea conținutului
declare (**encoding**='ISO-8859-1');

// verificare strictă a tipurilor de date la PHP 7
declare (**strict_types**=1);

php: caracterizare

Pentru creșterea performanței,
se poate adopta compilarea *just-in-time*

HHVM – HipHop Virtual Machine (Facebook)
utilizată de Baidu, Box, Etsy, Facebook, Wikipedia,...

www.hhvm.com

php: caracterizare

Interacțiunea cu utilizatorul:

preluarea valorilor câmpurilor formularelor Web

cookie-uri

sesiuni

autentificarea utilizatorului

acces la variabile globale create „din zbor”

php: caracterizare

Facilități pentru tehnologiile Web:

prelucrare de URL-uri

suport pentru HTTP – inclusiv cURL

caching via memcached

dezvoltare de servicii Web prin SOAP și REST

...și altele

php: caracterizare

Suport pentru acces la baze de date:

la nivel abstract

DBA (*DataBase Abstraction layer*)

iODBC (*Independent Open DataBase Connectivity*)

PDO (*PHP Data Objects*)

www.phptherightway.com/#databases_abstraction_layers

php: caracterizare

Suport pentru acces la baze de date:

specific unui server de baze de date

relațional: DB2, MySQL, Oracle, PostgreSQL, SQLite,...

bazat pe NoSQL – *e.g.*, MongoDB

a se parcurge <http://www.phptherightway.com/#databases>

php: caracterizare

Prelucrarea conținutului resurselor:

fișiere audio – via biblioteci: ktaglib, oggvorbis etc.

arhive de tip bzip2, LZF, RAR, ZIP, ZLIB

documente PDF

imagini – cu biblioteci: Cairo, GD, EXIF, ImageMagick

fișiere în format JSON

documente XML – creare, procesare, validare etc.

cărți de credit

...

php: caracterizare

Suport pentru resurse de sistem + Internet:

sisteme de fişiere, inclusiv FTP
processe – cu libevent, pthreads, Gearman,...
răspuns la evenimente – via Event
socket-uri
poştă electronică – *e.g.*, IMAP, POP3
...şi multe altele

PHP ca limbaj de programare procedurală

php: tipuri de date – scalare

boolean

TRUE sau FALSE

php: tipuri de date – scalare

integer

valori întregi specificate
în baza 10 sau 16 (hexa), 8 (octal), 2 (binar)

php: tipuri de date – scalare

float

numere reale

reprezentate uzual conform IEEE 754 (dublă precizie)

php: tipuri de date – scalare

float

numere reale

reprezentate uzual conform IEEE 754 (dublă precizie)

valoare specială – constanta NAN (*not a number*)

funcții predefinite utile: `is_nan()`, `is_finite()`, `is_infinite()`

php: tipuri de date – scalare

string

șiruri de caractere ASCII
(nu există suport nativ pentru Unicode)

ca la C, pot fi folosite caractere *escape* precum

`\n` `\r` `\t` `\e` `\\` `\$` `\"`

php: tipuri de date – scalare

string

delimitatori uzuali:
" sau '

un șir nu poate avea mai mult de 2 GB

php: tipuri de date – compuse

array

asociere între valori (de orice tip)
și chei (de tip integer sau string)

nu există o distincție clară
între tablouri indexate și cele asociative

php: tipuri de date – compuse

array

// un tablou indexat (vector de valori)

```
$cadouri = array ("trotineta", "laptop", "minge", "topor");
```

// un tablou asociativ – perechi (cheie, valoare)

```
array ( "nume" => "Tux", "dimensiune" => 17, "oferta" => TRUE );
```

// sintaxa simplificată (pentru PHP 5.4+)

```
[ "nume" => "Tux", "dimensiune" => 17, "oferta" => TRUE ];
```

php: tipuri de date – compuse

object

instanță a unei clase

creat cu operatorul **new**

php: tipuri de date – speciale

resource

semnifică o referință la o resursă externă

o resursă e creată de funcții specifice

e.g., resursa de tip *stream* inițiată de funcția `fopen()`
și folosită de funcțiile `fread()`, `feof()`, `fgets()` etc.

funcții predefinite: `is_resource()`, `get_resource_type()`

detalii la <http://www.php.net/manual/en/resource.php>

php: tipuri de date – speciale

null

specifică valoarea NULL
reprezentând o variabilă care nu are valoare

funcții utile: `is_null()` și `unset()`

php: variabile

Variabile create „din zbor”
tipul e determinat pe baza contextului

convertirea automată a tipului (*type casting*)
e similară celei de la limbajul C

```
$ani = 21;  
$conectat = TRUE;  
$prefer["culoare"] = "gri";
```

```
/* o variabilă obișnuită */  
# una de tip Boolean  
// un tablou asociativ
```

php: variabile

Variabile create „din zbor”
tipul e determinat pe baza contextului

funcții predefinite utile:
var_dump(), settype(),
is_bool(), is_int(), is_float(), is_array(), is_string(),
is_scalar(), is_numeric() etc.

php: variabile

Vizibilitatea variabilelor (*scope*)

pentru a putea fi folosite în întreg programul,
variabilele trebuie declarate ca fiind globale

```
$scor = 33;
```

```
function oferaScor () {  
    echo "Scor curent: " . $scor;  
}
```

```
oferaScor();
```

- **Undefined variable:**
scor in prog.php on line 4

```
$scor = 33;
```

```
function oferaScor () {  
    global $scor;  
    echo "Scor curent: " . $scor;  
    // similar cu $GLOBALS["scor"]  
}
```

```
oferaScor();
```

- **Scor curent: 33**

php: variabile

Vizibilitatea variabilelor (*scope*)

o variabilă poate fi declarată ca fiind **statică**

există doar în domeniul de vizibilitate local
(*e.g.*, în cadrul unei funcții), dar nu-și pierde valoarea
atunci când execuția programului părăsește
acel domeniu de vizibilitate

php: variabile predefinite

Variabile disponibile în întreg programul
(*superglobals*)

\$GLOBALS []

tablou asociativ ce conține referințe
la toate variabilele definite global

php: variabile predefinite

Variabile disponibile în întreg programul
(*superglobals*)

`$_SERVER []`
`$_GET []` `$_POST []` `$_FILES []` `$_REQUEST []`
`$_SESSION []`
`$php_errormsg`
`$argc` `$argv`

...

php: constante

Specificate cu **define ()**

sunt disponibile la nivel global în program

```
define ("DIMENS_MIN", "15");
```


php: constante predefinite

Exemplificări:

PHP_VERSION

PHP_OS

PHP_EOL

PHP_INT_MAX

PHP_INT_SIZE

TRUE

FALSE

NULL

php: constante predefinite

Controlul manierei de raportare a erorilor:

E_ERROR	erori fatale (execuția <i>script</i> -ului e oprită)
E_WARNING	avertismente
E_PARSE	erori de procesare a codului (<i>parsing</i>)
E_NOTICE	notificări în timpul rulării
E_STRICT	sugestii privind îmbunătățirea codului
E_DEPRECATED	notificări despre aspecte demodate

www.php.net/manual/en/errorfunc.constants.php
www.phptherightway.com/#errors_and_exceptions

php: constante predefinite

Mediul de execuție oferă acces la constante „magice” ale căror valori pot fi folosite în cadrul programului

__LINE__
__FILE__
__DIR__
__FUNCTION__
__CLASS__
__TRAIT__
__METHOD__
__NAMESPACE__

php: operatori

Majoritatea, similari celor din limbajul C

aritmetici: **+** **-** ***** **/** **%** **++** **--**

asignare a valorii: **=** și **=>** (pentru tablouri)

asignare prin referință: **=&**

pe biți: **&** **|** **^** **<<** **>>**

comparații: **==** **===** **!=** **<>** **!==** **<** **>** **<=** **>=** **?:** **??** **<=>**

control al raportării erorilor: **@**

logici: **and** **or** **xor** **!** **&&** **||**

șiruri de caractere (concatenare) – ca la Perl: **..=**

php: operatori

În PHP 7, se pot folosi și noii operatori:

<=> (*spaceship*)

compararea a două expresii (de tip scalar),
întorcând -1, 0 sau 1

echo 15.5 <=> 15.5;	// 0 (egalitate)
echo 15.5 <=> 16.5;	// -1 (mai mic)
echo 17.5 <=> 15.5;	// 1 (mai mare)

php: operatori

În PHP 7, se pot folosi și noii operatori:

?? (*null coalescing*)

oferă valoarea primului operand dacă există și nu e NULL,
altfel întoarce valoarea celui de-al doilea operand

```
// folosim ca nume de utilizator valoarea furnizată în formular  
// (preluată prin GET sau POST); dacă nu există, va fi 'tux'  
$username = $_GET['user'] ?? $_POST['user'] ?? 'tux';
```

php: structuri de control

if, switch, while, do, for, break, continue
asemănătoare celor din C

```
if (!$nume) {  
    echo ("Nu este precizat corect numele...");  
} else {  
    echo ("Bine ai venit, " . $nume . "!\n");  
}
```

php: exemplu

```
<?php
// umplem un tablou cu valori de la 1 la 10
for ($contor = 1; $contor <= 10; $contor++) {
    $valori[$contor] = $contor;
}
// realizăm suma valorilor
$suma = 0;
foreach ($valori as $element)
    $suma += $element;
/* afișăm suma obținută la ieșirea standard
   pentru a fi trimisă clientului Web */
echo("<p>Suma de la 1 la 10 este <strong>" . $suma . "</strong>.</p>");
?>
```

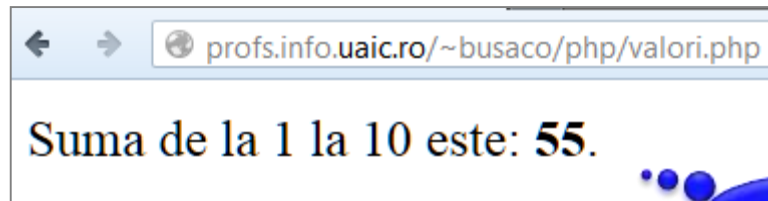

Invocarea (rularea) programului PHP direct
din linia de comandă:

salvăm codul într-un fișier text – **valori.php**
apelăm interpretorul PHP din linia de comandă

php valori.php

<p>Suma de la 1 la 10 este 55.</p>

Invocarea (rularea) programului PHP
la nivelul serverului Web:
plasăm fișierul sursă – cu drepturi de citire și execuție
în navigator, indicăm URL-ul către program
pentru a-l invoca via metoda GET a protocolului HTTP



rezultatul generat
de *script*

[edit](#) [fork](#) [download](#)

```
1.  <?php
2.      // umplem un tablou cu valori de la 1 la 10
3.      for ($contor = 1; $contor <= 10; $contor++) {
4.          $valori[$contor] = $contor;
5.      }
6.      // realizăm suma valorilor
7.      $suma = 0;
8.      foreach ($valori as $element)
9.          $suma += $element;
10.     /* afișăm suma obținută la ieșirea standard
11.        pentru a fi trimisă clientului Web */
12.     echo ("<p>Suma de la 1 la 10 este <strong>" . $suma . "</strong>.</p>");
13.  ?>
14.
```

Success

 stdin

Standard input is empty

 stdout

<p>Suma de la 1 la 10 este 55.</p>

recurgerea la un mediu
de execuție *online*
<http://ideone.com/>

php: structuri de control

Includerea de cod-sursă din alte fișiere
(suport pentru modularizare)

include

caută fișierul sursă în directoarele predefinite specificate
via **include_path** și-l evaluează

dacă fișierul nu există, se generează un avertisment

include_once – pentru a-l include o singură dată

php: structuri de control

Includerea de cod-sursă din alte fișiere
(suport pentru modularizare)

require

caută fișierul sursă în directoarele predefinite specificate
via **include_path** și-l evaluează


dacă fișierul nu există, se emite o eroare fatală

require_once – pentru a-l include o singură dată

php: funcții

Funcții definite de utilizator:

```
function trimiteMesaj ($exped="", $dest="", $subiect="Web") {  
    // corp...  
}
```



parametri cu
valori implicite

<http://php.net/manual/en/language.functions.php>

```
define ('MAX', 10);           // numărul maxim de valori

function patrat ($numar) {    // funcția de ridicare la pătrat
    return $numar * $numar;
}

$numumar = 0;
while ($numar < MAX) {
    $numar++;                 // incrementăm numărul

    if ($numar % 2)           // e număr impar...
        continue;            // continuăm cu următoarea iterație

    // e număr par, deci afișăm pătratul lui
    echo "$numar la pătrat este " . patrat ($numar) . "\n";
} // final de while
```

php: funcții

Funcții definite de utilizator:

numele funcțiilor sunt considerate *case-insensitive*

parametrii pot fi dați prin referință – prefixați de &

la PHP 5.6+, numărul variabil de parametri e indicat de ...

<http://php.net/manual/en/functions.arguments.php>


```
<?php
```

```
declare (strict_types=1);
```

```
// argumentele trebuie să fie întregi, valoarea oferită trebuie să fie de tip int
```

```
function aduna (int ...$numere): int {
```

```
    $suma = 0;
```

```
    foreach ($numere as $numar) {
```

```
        $suma += $numar;
```

```
    }
```

```
    return $suma;
```

```
}
```

```
echo aduna (7, 3, 74, 1);
```

```
echo aduna (pi (), '?');
```

```
?>
```

85

Fatal error: Uncaught TypeError:
Argument 1 passed to aduna() must
be of the type integer, float given
Next TypeError: Argument 2 passed
to aduna() must be of the type
integer, string given

în PHP 7, se poate preciza și tipul de date
pentru fiecare argument + valoarea întoarsă de funcție
(*scalar type declarations*)

php: funcții

Începând cu PHP 5.3,
pot fi specificate și funcții anonime
► programare funcțională (*e.g., closures*)

```
$saluta = function ($nume) {           // variabilă de tip funcție  
    printf ("Salut %s...\n", $nume);  
};
```

```
$saluta ('lumea');  
$saluta ('Tuxy');
```

vezi www.phptherightway.com/pages/Functional-Programming.html

php: funcții predefinite

matematice & de conversie
de manipulare a șirurilor de caractere
de prelucrare a tablourilor
de acces la resurse și de lucru cu fișiere
de manipulare a bazelor de date
privitoare la conexiunile de rețea
pentru accesarea resurselor XML, PDF, JPEG,...
specifice sistemului de operare
generale

detalii la <http://php.net/manual/en/funcref.php>

php: funcții predefinite

Matematică:

abs(), mod(), fmod()
ceil(), floor(), round(), max(), min()
exp(), log10(), log()
pow(), sqrt()
sin(), cos(), tan(), asin(), ..., sinh(), ..., pi()
rand(), srand()
bindec(), octdec(), dechex(),..., base_convert()
is_finite(), is_infinite(), is_nan()

vezi și <http://php.net/manual/en/refs.math.php>

php: funcții predefinite

Șiruri de caractere:

echo(), print(), printf(), sprintf() etc.
strlen(), chr(), ord(), substr(), strstr(), strpos(),...
strcmp(), strcasecmp(), strnatcmp() etc.
strcat(), str_replace(), str_ireplace(), strrev() etc.
trim(), ltrim(), rtrim()
explode(), implode(), split(), join(), strtok()

detalii vizând procesarea textelor:

<http://php.net/manual/en/refs.basic.text.php>

php: funcții predefinite

Expresii regulate:

conform standardului POSIX

`ereg()`, `ereg_replace()`, `split()` etc.

compatibile cu cele din Perl – PCRE: <http://www.pcre.org/>

`preg_filter()`, `preg_grep()`, `preg_match()`, `preg_split()`,...

php: funcții predefinite

Tablouri:

`array_count_values()`, `array_search()`, `array_filter()`,
`array_slice()`, `array_chunk()`
`array_fill()`, `array_combine()`, `array_shift()`,
`array_reverse()`, `array_multisort()`, `array_sum()`,...
`array_merge()`, `array_intersect()`, `array_diff()`
`array_keys()`, `array_key_exists()`
`array_push()`, `array_pop()`
`array_map()`, `array_reduce()`

<http://php.net/manual/en/book.array.php>

/* filtrarea unor valori dintr-un tablou
pe baza unei funcții specificate de programator */

```
function valoare_mai_mica_decat ($numar) {  
    // întoarce o expresie de tip funcție  
    return function ($element) use ($numar) { // abordare funcțională  
        return $element < $numar;  
    };  
}
```

```
$punctaje = array (7, 8, 9, 10, 5, 3, 10, 6, 4);  
// folosim funcția predefinită array_filter() asupra tabloului cu punctaje  
// pentru a obține valorile mai mici decât o valoare dată (aici: 7)  
$valori = array_filter ($punctaje, valoare_mai_mica_decat (7));  
  
print_r ($valori); // obținem: Array ( [4] => 5 [5] => 3 [7] => 6 [8] => 4 )
```

a se studia și <https://wiki.php.net/rfc/closures>

php: funcții predefinite

Manipulare a caracterelor:

`ctype_digit()`, `ctype_xdigit()`, `ctype_print()`,
`ctype_punct()`, `ctype_space()`,...

`ctype_alpha()`, `ctype_alnum()`, `ctype_lower()`,
`ctype_upper()`

php: funcții predefinite

Dată & timp:

getdate(), localtime(), gettimeofday(), time() etc.
date(), idate(), gmdate(),...
checkdate()
strftime(), strtotime()

vezi și extensiile **Calendar**, **DateTime**, **HRTime**

php: funcții predefinite

Variabile PHP:

`empty()`, `isset()`, `unset()`
`strval()`, `print_r()`, `var_dump()`
`serialize()`, `unserialize()`

a se consulta și <http://php.net/manual/en/book.var.php>

php: funcții predefinite

Fișiere și directoare:

folosind tipul de date FILE – ca la limbajul C:

`fopen()`, `fread()`, `fscanf()`, `fgets()`, `fwrite()`, `fprintf()`,
`fseek()`, `ftell()`, `feof()`, `fclose()`, `ftruncate()`, `fstat()`,...
`file()`, `copy()`, `rename()`, `delete()`,
`move_uploaded_file()`, `tmpfile()`
`file_exists()`, `filesize()`, `filetype()`, `fileperms()`,..., `stat()`
`is_dir()`, `is_file()`, `is_readable()`, `is_writeable()`,...
`chdir()`, `mkdir()`, `rmdir()`
`disk_free_space()`, `disk_total_space()`

de studiat și <http://php.net/manual/en/refs.fileprocess.file.php>

php: funcții predefinite

URL-uri:

`urldecode()`, `urlencode()`, `parse_url()`
`base64_decode()`, `base64_encode()`
`http_build_query()`

php: funcții predefinite

Prelucrarea resurselor Web (HTML, JSON):

`nl2br()`, `htmlentities()`, `htmlspecialchars()`, `strip_tags()`

`get_browser()`, `show_source()`, `highlight_string()`,...

`json_encode()`, `json_decode()`, `json_last_error()`

php: funcții predefinite

Alte funcții utile:

`die()`, `eval()`, `exit()`, `sleep()`, `usleep()`, `time_sleep_until()`

`uniqid()`, `sys_getloadavg()`

`php_info()`, `php_check_syntax()`

php: alte facilități

SPL (*Standard PHP Library*)

acces la maniere standard de prelucrare a datelor

structuri de date definite:

SplStack, SplQueue, SplHeap, SplPriorityQueue,...

iteratori:

ArrayIterator, FilesystemIterator, RegexIterator etc.

www.php.net/spl

www.phptherightway.com/#standard_php_library

php: alte facilități

Rularea din linie de comandă – **PHP CLI**
sau ca modul Apache

FPM (*FastCGI Process Manager*)
soluție vizând performanța

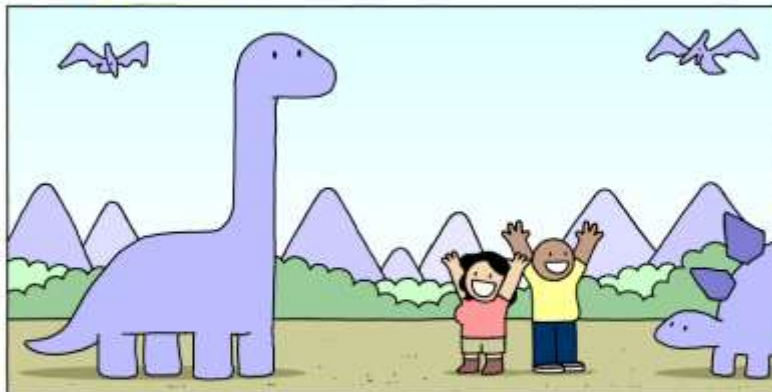
de asemenea, PHP 5.4+ oferă un server Web incorporat
invocat prin **php -S localhost:8000 -t phpwebapp/**
<http://php.net/features.commandline.webserver>

php: alte facilități

Inter-conectivitatea cu alte tehnologii/platforme

exemple: Java, JavaScript, Lua, .NET

(în loc de) pauză



Care-i suportul oferit de PHP
pentru programarea obiectuală?

php: clase

Suport pentru definirea claselor via **class**
și de instanțiere prin operatorul **new**

obiectele sunt tratate similar referințelor
(o variabilă de tip obiect conține o referință la un obiect
și nu o copie a lui)

<http://php.net/manual/en/oop5.intro.php>
detalii la <http://php.net/language.oop5>

programare obiectuală – încapsularea

```
class Student { // specificarea unei clase
  // proprietăți (date-membre)
  private $an;
  private $email;
  public $nume;
  // metode publice
  public function seteazaAn ($unAn) {
    $this->an = $unAn;
  }
  public function furnizeazaAn () {
    return $this->an;
  }
}
```

\$this este o pseudo-variabilă
specificând o referință la obiectul curent

programare obiectuală – încapsularea

```
class Student { // specificarea unei clase
// proprietăți (date-membre)
private $an;
private $email;
public $nume;
// metode publice
public function seteazaAn ($unAn) {
    $this->an = $unAn;
}
public function furnizeazaAn () {
    return $this->an;
}
}
```

```
// instanțierea unui obiect
$stud = new Student ();
$stud->seteazaAn (2);
$stud->nume = 'Tux';
print_r ($stud);
```

► Student Object

```
(
    [an:Student:private] => 2
    [nume] => Tux
    [email:Student:private] =>
)
```


php: clase

Ca la C++, membrii – proprietăți sau metode –
pot fi declarați ca fiind

publici (**public**)

privați (**private**)

protejați (**protected**)

programare obiectuală – moștenirea

```
class StudentDestept extends Student {  
    private $note;    // notele obținute (proprietate)  
  
    public function seteazaNote ($n) {  
        $this->note = (array) $n;  
    }  
    public function furnizeazaNote () {  
        return (array) $this->note;  
    }  
}  
  
$altStud = new StudentDestept ();  
// apel de metodă din clasa de bază  
$altStud->seteazaAn (2);  
// apel de metodă din clasa derivată  
$altStud->seteazaNote (  
    ['TW' => 10, 'IP' => 9]  
);
```

programare obiectuală – moștenirea

```
class StudentDestept extends Student {  
    private $note;    // notele obținute (proprietate)
```

```
    public function seteazaNote ($n) {  
        $this->note = (array) $n;
```

```
    }  
    public function furnizeazaNote () {  
        return (array) $this->note;
```

```
    }  
}
```

```
$altStud = new StudentDestept ();  
// apel de metodă din clasa de bază  
$altStud->seteazaAn (2);  
// apel de metodă din clasa derivată  
$altStud->seteazaNote (  
    ['TW' => 10, 'IP' => 9]  
);
```

```
print_r ($altStud);
```

► StudentDestept Object

```
(  
    [note:StudentDestept:private]  
    => Array  
    (  
        [TW] => 10  
        [IP] => 9  
    )  
  
    [an:Student:private] => 2  
    [nume] =>  
    [email:Student:private] =>  
)
```

php: clase

Metode speciale:

constructorii sunt numiți **__construct()**

destructorii sunt denumiți **__destruct()**

php: clase

Accesarea proprietăților/metodelor
statice, constante sau suprascrise



scope resolution operator (Paamayim Nekudotayim)

www.php.net/manual/en/language.oop5.paamayim-nekudotayim.php

php: clase

Accesarea proprietăților/metodelor
statice, constante sau suprascrise



self – clasa curentă
parent – clasa părinte

php: clase

Proprietățile sau metodele declarate cu **static** pot fi accesate fără a fi nevoie de instanțierea clasei

pentru exemple, a se vizita

www.php.net/manual/en/language.oop5.static.php

php: clase

Se permit clase/metode abstracte declarate cu **abstract**

clasele abstracte nu pot fi instanțiate

orice clasă având măcar o metodă abstractă
este considerată abstractă

metodele abstracte trebuie implementate
în clasa copil (specificată cu **extends**) a clasei abstracte

php: interfețe

Specificarea metodelor ce vor fi ulterior implementate de o clasă (ca la Java)

// interfața privind o machetă de vizualizare (*template*)

```
interface iMacheta {  
    // setează o variabilă ce va fi substituită  
    // cu valoarea ei în cadrul machetei  
    public function setVar ($nume, $var);  
    // furnizează reprezentarea machetei  
    public function oferaReprez ($macheta);  
}
```

amănunte la <http://php.net/manual/en/language.oop5.interfaces.php>

```
// clasa implementând interfața
class Macheta implements iMacheta {
    // tablou asociativ cu variabilele ce trebuie înlocuite cu valorile lor
    private $variabile = array ();

    public function setVar ($nume, $var) {
        $this->variabile[$nume] = $var;
    }

    public function oferaReprez ($macheta) {
        foreach ($this->variabile as $nume => $val) {
            // substituim în machetă numele variabilelor cu valorile lor
            $macheta = str_replace ('{' . $nume . '}', $val, $macheta);
        }
        return $macheta;
    }
}
```

aspecte mai avansate: www.phptherightway.com/#templating

php: interfețe & clase predefinite

Traversable
Iterator
IteratorAggregate
Throwable
ArrayAccess
Serializable
Closure
Generator

<http://www.php.net/manual/en/reserved.interfaces.php>

php: interfețe & clase predefinite

// interfața Iterator

Iterator extends **Traversable** {

// metode ce trebuie scrise de programator

// în clasa ce implementează interfața

abstract public mixed **current** (void)

abstract public scalar **key** (void)

abstract public void **next** (void)

abstract public void **rewind** (void)

abstract public boolean **valid** (void)

}

php: trăsături de clasă (*traits*)

Trait

concept preluat de la limbajul Self, oferit de PHP 5.4+

colecție de metode ce pot fi refolosite în cadrul altor clase

www.php.net/manual/en/language.oop5.traits.php

php: trăsături de clasă (*traits*)

Trait

considerat ca șablon (*template* C++) al unei clase
față de interfețe, oferă implementări ale metodelor,
nu doar signaturile lor

// trăsături (comportamente) ce vor fi asociate unor figuri geometrice 2D

```
trait Rotire {  
  public function roteste ($unghi) {           // implementează rotirea  
  }  
}
```

```
trait Mutare {  
  public function mutaLa ($x, $y) {           // mută la alte coordonate  
  }  
}
```

```
trait Colorare {  
  public function coloreaza ($culoare) { // realizează colorarea  
  }  
}
```

```
abstract class Figura { // clasa figurilor geometrice
    public function deseneaza() {
        echo ('Am desenat ' . get_class());
    }
}
```

```
class Dreptunghi extends Figura {
    use Colorare, Mutare, Rotire;

    public function transforma () {
    }
}
```

// folosește trăsăturile dorite
// poate fi colorat, mutat, rotit

// în plus, o transformare specifică

```
final class Cerc extends Figura {
    use Mutare, Colorare;

    const PI = 3.1415265;

    public function calculeazaArie () {
    }
}
```

// clasa Cerc nu mai poate fi extinsă
// un cerc poate fi mutat și colorat

php: trăsături de clasă (*traits*)

```
// instanțiem 2 figuri: un cerc și un dreptunghi  
$unCerc = new Cerc ();  
$unDreptunghi = new Dreptunghi ();  
$unCerc->deseneaza ();  
$unCerc->roteste (); // va cauza emiterea unei erori  
$unDreptunghi->deseneaza ();
```

► Am desenat Cerc

**PHP Fatal error: Call to undefined method Cerc::roteste()
in /home/dMdWgn/prog.php on line 47**

php: proprietăți speciale

O clasă are asociate proprietăți speciale („magice”) ce pot fi suprascrise

`__construct ()`

`__destruct ()`

`__toString ()`

`__get ()`

`__set ()`

altele la www.php.net/manual/en/language.oop5.magic.php

php: obiecte

Obiectele pot fi „clonate” via **clone**

Obiectele se pot compara folosind operatorul **===**

php: obiecte

Funcții de manipulare a claselor și obiectelor

- `get_class()` va returna numele unui obiect, instanță a unei clase
- `get_parent_class()` furnizează clasa părinte din care provine un anumit obiect
- `method_exists()` testează dacă există o metodă pentru un anumit obiect specificat
- `class_exists()` testează existența unei clase
- `is_subclass_of()` determină dacă există o relație de moștenire dintre două clase

php: excepții

Similare celor din Java

try, catch, throw
clasa Exception

detalii la www.php.net/manual/en/language.exceptions.php

php: spații de nume

Folosite pentru evitarea coliziunilor de nume și pentru efectuarea de *alias*-uri

declarare cu **namespace** (prima linie de program)

exemplu: **namespace Facebook;** // Facebook SDK for PHP

de studiat și www.phptherightway.com/#namespaces

php: spații de nume

Folosite pentru evitarea coliziunilor de nume și pentru efectuarea de *alias*-uri

același spațiu de nume poate fi definit în fișiere multiple

pot exista ierarhii de (sub-)spații de nume

```
namespace Proiect\Modul\Submodul;  
  
class GenSVG { ... };
```

referire cu
Proiect\Modul\Submodul\GenSVG

php: spații de nume

Folosite pentru evitarea coliziunilor de nume și pentru efectuarea de *alias*-uri

utilizare prin **use** (eventual specificând un *alias*)

```
use Proiect\Modul\Submodul;
```

exemple concrete:

```
use Facebook\Authentication\AccessToken;
```

```
use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;
```


php: spații de nume

Folosite pentru evitarea coliziunilor de nume și pentru efectuarea de *alias*-uri

construcțiile care nu aparțin niciunui spațiu de nume definit sunt considerate ale spațiului de nume global

```
namespace ProiectWeb;  
function fopen () {      // specificarea unei funcții proprii  
...  
    $fisier = \fopen (...); // apel de funcție predefinită (din spațiul global)  
}
```

Care sunt facilitățile
vizând interacțiunea Web?

php: interacțiune web

Datele transmise de client (*browser*) se regăsesc în tablouri asociative predefinite (și globale):

\$_GET[] – datele transmise prin GET

\$_POST[] – datele transmise prin POST

\$_COOKIE[] – *cookie*-urile receptate

\$_REQUEST[] – datele primite de la client
(conținutul lui **\$_GET**, **\$_POST** și **\$_COOKIE**)

\$_SESSION[] – datele de tip sesiune

php: interacțiune web

Alte variabile globale utile:

`$_SERVER[]`

oferă informații privind serverul Web

`$_SERVER['PHP_SELF']` indică numele *script*-ului PHP

`$_SERVER['REQUEST_METHOD']`

`$_SERVER['HTTP_REFERER']`

`$_SERVER['HTTP_USER_AGENT']`

<http://www.php.net/manual/en/reserved.variables.server.php>


php: interacțiune web

Alte variabile globale utile:

\$_ENV[] – datele oferite de mediu (*environment*)

\$_FILES[] – datele despre fișierele primite prin *upload*

<http://www.php.net/manual/en/features.file-upload.php>




vezi exemplul din arhiva
aferentă prelegerii

```
<!-- un formular Web modelat în HTML -->  
<form action="afiseaza.php" method="post">  
    <input type="text" name="nume" />  
    <input type="text" name="varsta" />  
    <input type="submit" value="Trimite" />  
</form>
```

```
<!-- un formular Web modelat în HTML -->  
<form action="afiseaza.php" method="post">  
    <input type="text" name="nume" />  
    <input type="text" name="varsta" />  
    <input type="submit" value="Trimite" />  
</form>
```

```
<?php  
    // programul afiseaza.php invocat prin POST  
    if (!$_REQUEST["nume"])  
        afiseaza ("Nu ați specificat numele!", "eroare");  
    else  
        afiseaza ("Numele este" . $_REQUEST["nume"]);  
?>
```



fiecare nume de câmp din formular reprezintă
o cheie a tabloului asociativ **\$_REQUEST []**
(în funcție de metoda HTTP, poate fi regăsit în **\$_GET** sau **\$_POST**)

php: sesiuni web

Managementul sesiunilor se poate realiza via funcțiile `session_*`() sau recurgând la clasa `SessionHandler`

```
SessionHandler implements SessionHandlerInterface {  
    public bool  open ( string $save_path , string $session_name )  
    public string create_sid ( void )  
    public string read ( string $session_id )  
    public bool  write ( string $session_id , string $session_data )  
    public bool  gc ( int $maxlifetime )  
    public bool  destroy ( string $session_id )  
    public bool  close ( void )  
}
```

<http://php.net/manual/en/book.session.php>

Cum pot fi accesate bazele de date din PHP?

php: bd

Suport nativ pentru o multitudine
de servere/tehnologii de baze de date:

MongoDB – clasele **MongoDB MongoClient MongoClient**

MySQL / MariaDB – clasa **mysqli**

PostgreSQL – funcțiile **pg_***()

SQLite – clasa **SQLite3**

etc.

conexiunile pot fi persistente

php: bd – mysql

Funcții/metode pentru acces la MySQL/MariaDB

- conectare la server: **mysql_connect ()**, **mysql_pconnect ()**
- selectare (utilizare) bază de date: **mysql_select_db ()**
- execuția unei interogări: **mysql_query ()**
- raportare de erori: **mysql_errno ()**, **mysql_error ()**
- preluarea rezultatelor într-un tablou: **mysql_fetch_array ()**
- multe altele...

actualmente, **abordare depreciată** – eliminată în PHP 7

php: bd – extensia mysqli

Scop: acces usor și flexibil la MySQL/MariaDB
din programele PHP5+

facilitează mentenabilitatea codului

compatibilitate cu API-ul MySQL

alternativă la funcțiile mysql_*

documentații disponibile la www.php.net/mysqli

php: bd – extensia **mysqli**

Abordare procedurală sau orientată-obiect

Viteză mai mare de procesare + securitate

Metode importante:

inițierea unei conexiuni cu serverul MySQL – **mysqli ()**

interogări SQL – **query ()**, **prepare ()**, **execute ()**

procesarea răspunsului – **fetch ()**, **fetch_assoc ()**

închiderea conexiunii – **close ()**

etc.

php: bd – exemplu

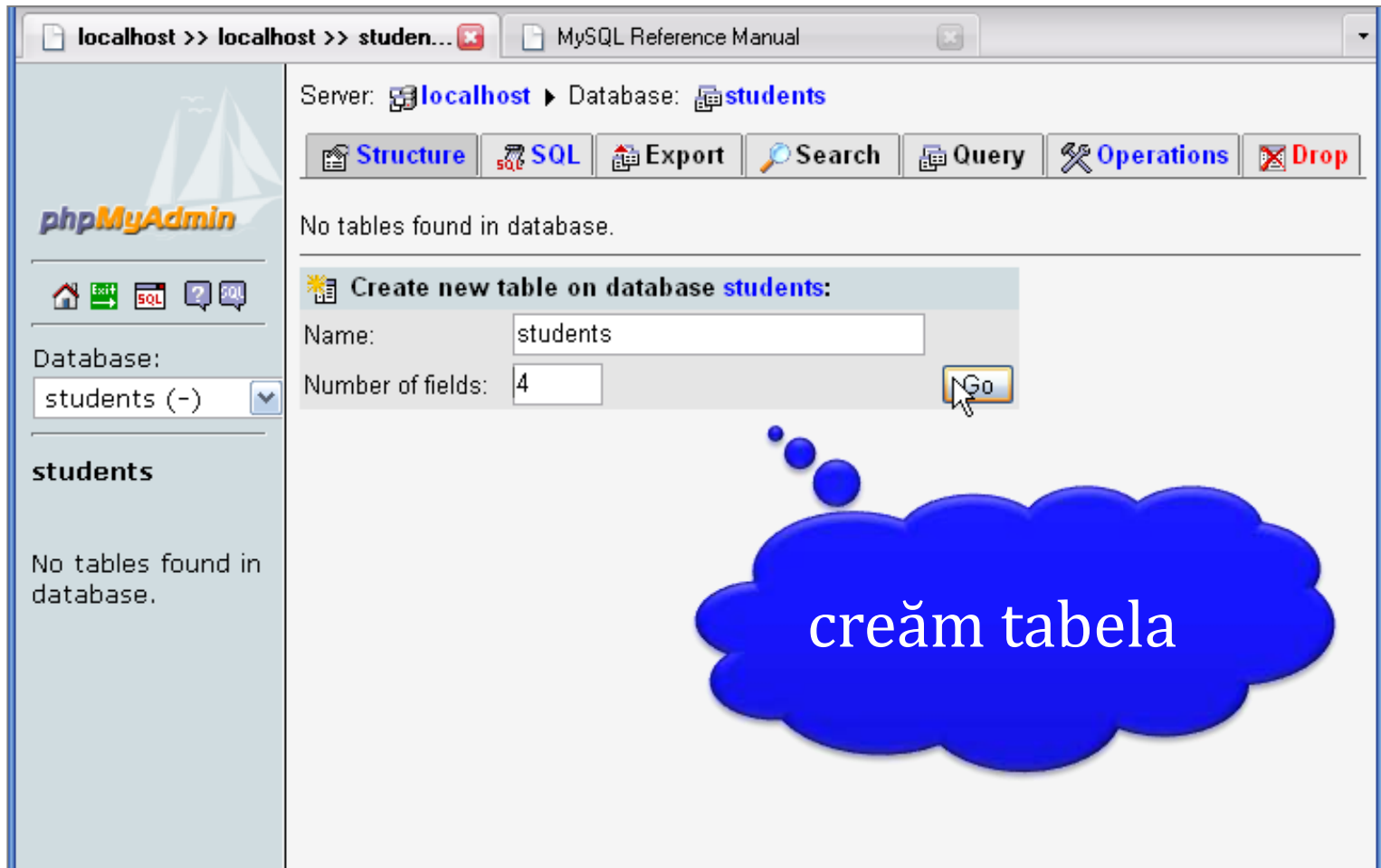
Pentru început, vom crea un cont MySQL care să asigure acces autentificat din programele PHP asupra bazei de date **students**:

```
(infoiasi)$ mysql -u root mysql
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE,
DROP ON students.* TO 'tux'@'localhost' IDENTIFIED BY
'p@rola' WITH GRANT OPTION;
Query OK, 0 rows affected (0.11 sec)
```

php: bd – exemplu

Folosind clientul `mysql` în linia de comandă sau `PHPMyAdmin`, creăm tabela `students` cu structura:

```
CREATE TABLE IF NOT EXISTS `students` (  
  `name` varchar(50) NOT NULL default "",  
  `year` enum('1','2','3') NOT NULL default '1',  
  `id` int(11) NOT NULL auto_increment,  
  `age` smallint(2) unsigned zerofill NOT NULL default '00',  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
AUTO_INCREMENT=1;
```



pentru administrare facilă, recurgem la instrumentul Web
phpMyAdmin – <https://www.phpmyadmin.net/>

phpMyAdmin

Server: localhost ▶ Database: students ▶ Table: students

Field	Type?	Length/Values*	Collation
name	VARCHAR	50	
year	ENUM	'1','2','3'	
id	BIGINT	11	
age	DOUBLE	2	UNSIGNED

Database: students (-)

students

No tables found in database.

Add 1 field

Go

Table comments:

Table type: MyISAM Collation:

* If field type is 'enum' or 'set', you must precede it with 'enum' or 'set' respectively.
** For default value, use 'DEFAULT'.

generăm structura

phpMyAdmin

Server: localhost ▶ Database: students ▶ Table: students

Structure Browse SQL Search Insert Export Operations

Table students has been created.


SQL query:

```
CREATE TABLE `students` (  
  `name` VARCHAR(50) NOT NULL ,  
  `year` ENUM('1', '2', '3') NOT NULL ,  
  `id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `age` SMALLINT(2) UNSIGNED ZEROFILL NOT NULL ,  
  PRIMARY KEY (`id`)  
) TYPE = MYISAM ;
```

[Edit] [Create PHP Code]

confirmarea creării

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	name	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	year	enum('1', '2', '3')	latin1_swedish_ci		No	1	
<input type="checkbox"/>	<u>id</u>	int(11)			No		auto_increment



Database:

students (1)

students

students

Field	Type	Function	Null	Value
name	varchar(50)			Cristian Vrabie
year	enum	--		<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3
id	int(11)			12341450890
age	smallint(2) unsigned zerofill			21

☒ Ignore

Field	Type	Function	Null	Value
name	varchar(50)			
year	enum	--		<input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3
id	int(11)			
age	smallint(2) unsigned zerofill			00

Insert as a new row -- And --

eventual, inserăm
o înregistrare

php: bd – extensia mysqli

```
// instanțiem obiectul mysqli
$mysql = new mysqli ('localhost', 'tux', 'p@rola', 'students');
if (mysqli_connect_errno ()) {
    die ('Conexiunea a eșuat...');
}

// formulăm o interogare și o executăm
if (!($rez = $mysql->query ('select name, year from students')) {
    die ('A survenit o eroare la interogare');
}
```

php: bd – extensia mysqli

```
// generăm o listă numerotată cu datele despre studenți  
// (cod HTML în stil spaghetti – practică nerecomandată)  
echo ('<ol>');  
while ($inreg = $rez->fetch_assoc ()) {  
    echo ('<li>Studentul ' . $inreg['name'] .  
        ' este în anul ' . $inreg['year'] . '</li>');  
}  
echo ('</ol>');  
  
// închidem conexiunea cu serverul MySQL  
$mysql->close ();
```

php: bd

În practică, se recurge la un strat de abstractizare a accesului la sistemul de stocare:

DBAL – DataBase Abstraction Layer

uzual, peste DBAL se va folosi o soluție (*i.e.* componentă, bibliotecă,...) de tip *ORM – Object-Relational Mapping*


exemple de instrumente PHP:


Doctrine – <http://www.doctrine-project.org/>

Propel – <http://propelorm.org/>

RedBeanPHP – <http://redbeanphp.com/>

Instrumente utile pentru dezvoltatorii Web?

pear  Register Login

Search for in the 

Main [Support](#) [Documentation](#) [Packages](#) [Package Proposals](#) [Developers](#) [Bugs](#)

[List Packages](#) [Search Packages](#) [Statistics](#) [Channels](#)

Packages (232) Filter by [PHP 4](#) **[PHP 5+](#)** [All PHP Versions](#)

Authentication 3 Auth_RADIUS, Auth_SASL2, OpenID	Configuration 1 Config_Lite	Console 4 Console_Color2, Console_CommandLine, Console_GetoptPlus
Database 16 DB_DataObject, DB_Sqlite_Tools, MDB2, MDB2_Driver_fbsql, ...	Date and Time 9 Calendar, Date_Holidays, Date_Holidays_Australia, Date_Holidays_Austria, ...	Encryption 6 Crypt_CHAP, Crypt_DiffieHellman, Crypt_GPG, Crypt_HMAC2, ...
Event 1 Event_SignalEmitter	File Formats 14 File_Bittorrent2, File_Cabinet, File_CSV_DataSource, File_DeliciousLibrary, ...	File System 1 File_Mogile
Gtk2 Components 7 Gtk2_EntryDialog, Gtk2_ExceptionDump, Gtk2_FileDrop, Gtk2_IndexedComboBox, ...	HTML 8 HTML_BBCodeParser2, HTML_Common2, HTML_Entities, HTML_QuickForm2, ...	HTTP 6 HTTP2, HTTP_FloodControl, HTTP_Header2, HTTP_OAuth, ...
Images 9 Image_3D, Image_Barcode2, Image_Color2, Image_Graph, ...	Internationalization 2 I18N_UnicodeNormalizer, I18N_UnicodeString	Logging 1 Log

PEAR (*PHP Extension and Application Repository*)
 module ce extind funcționalitățile PHP: <http://pear.php.net/>

- [Home](#)
- [News](#)

Documentation:

- [Support](#)

Downloads:

- **Browse Packages**
- [Search Packages](#)
- [Download Statistics](#)

Contents of :: [Top Level](#)

[Authentication](#) (5)

[courierauth](#), [krb5](#), [PAM](#), [radius](#) »

[Caching](#) (12)

[APC](#), [APCu](#), [chdb](#), [coherence](#) »

[Console](#) (5)

[ncurses](#), [newt](#), [termbox](#), [tvision](#) »

[Date and Time](#) (3)

[date](#) [time](#), [hrtime](#), [timezonedb](#)

[Event](#) (4)

[eio](#), [ev](#), [libevent](#), [swoole](#) »

[File System](#) (6)

[dazuko](#), [fuse](#), [inotify](#), [moglefs](#) »

[HTML](#) (2)

[html](#) [parse](#), [tidy](#)

[Images](#) (12)

[cairo](#), [cairo](#) [wrapper](#), [framegrab](#), [FreeImage](#) »

[Languages](#) (5)

[lua](#), [perl](#), [python](#), [spidermonkey](#) »

[Mail](#) (3)

[mailparse](#), [POP3](#), [vpopmail](#)

[Multimedia](#) (5)

[Audio](#), [ecasound](#), [ming](#), [opengl](#) »

[Numbers](#) (1)

[Bitset](#)

[Benchmarking](#) (3)

[DTrace](#), [included](#), [memprof](#)

[Configuration](#) (2)

[augeas](#), [zookeeper](#)

[Database](#) (54)

[couchbase](#), [CUBRID](#), [daffodildb](#), [dbase](#) »

[Encryption](#) (4)

[gnupg](#), [libsodium](#), [mcrypt](#) [filter](#), [script](#) »

[File Formats](#) (14)

[archive](#), [bz2](#), [cpdf](#), [Fileinfo](#) »

[GUI](#) (5)

[php](#) [xcb](#), [tk](#), [win32std](#), [WinBinder](#) »

[HTTP](#) (5)

[apfd](#), [json](#) [post](#), [pecl](#) [http](#), [txforward](#) »

[Internationalization](#) (8)

[cld](#), [fribidi](#), [gender](#), [idn](#) »

[Logging](#) (1)

[SeasLog](#)

[Math](#) (4)

[big](#) [int](#), [lapack](#), [stats](#), [trader](#) »

[Networking](#) (33)

[amqp](#), [apn](#), [ares](#), [cvsclient](#) »

[Payment](#) (6)

[coin](#) [acceptor](#), [cybercash](#), [cybermut](#), [mcve](#) »

Top Level

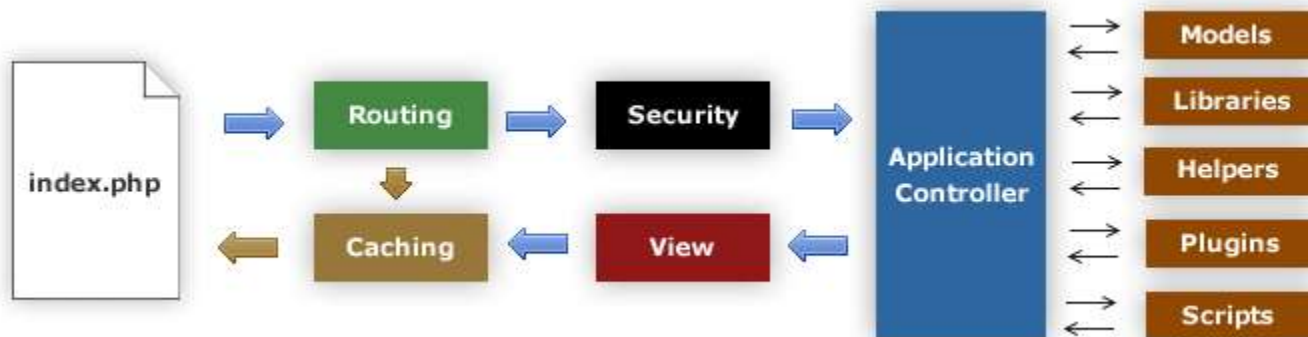
PECL (*PHP Extension Community Library*)

extensii oferite de terți: <http://pecl.php.net/>

instrumente: *framework-uri*

Facilități:

MV*, șabloane de proiectare,
acces la baze de date (ORM, DAO, ActiveRecord,...),
validare și filtrare a datelor de intrare, autentificare,
controlul accesului, management de sesiuni, *caching*,
transfer asincron de date (Ajax, WebSocket), *templating*,
suport pentru servicii Web și API-uri REST, module etc.



instrumente: *framework-uri*

CakePHP – <http://cakephp.org/>

CodeIgniter – <http://www.codeigniter.com/>

Laravel – <http://laravel.com/>

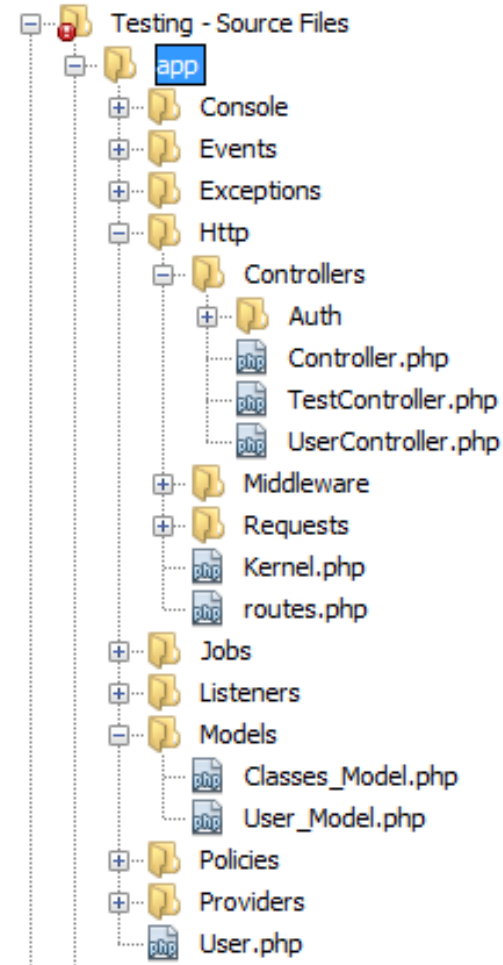
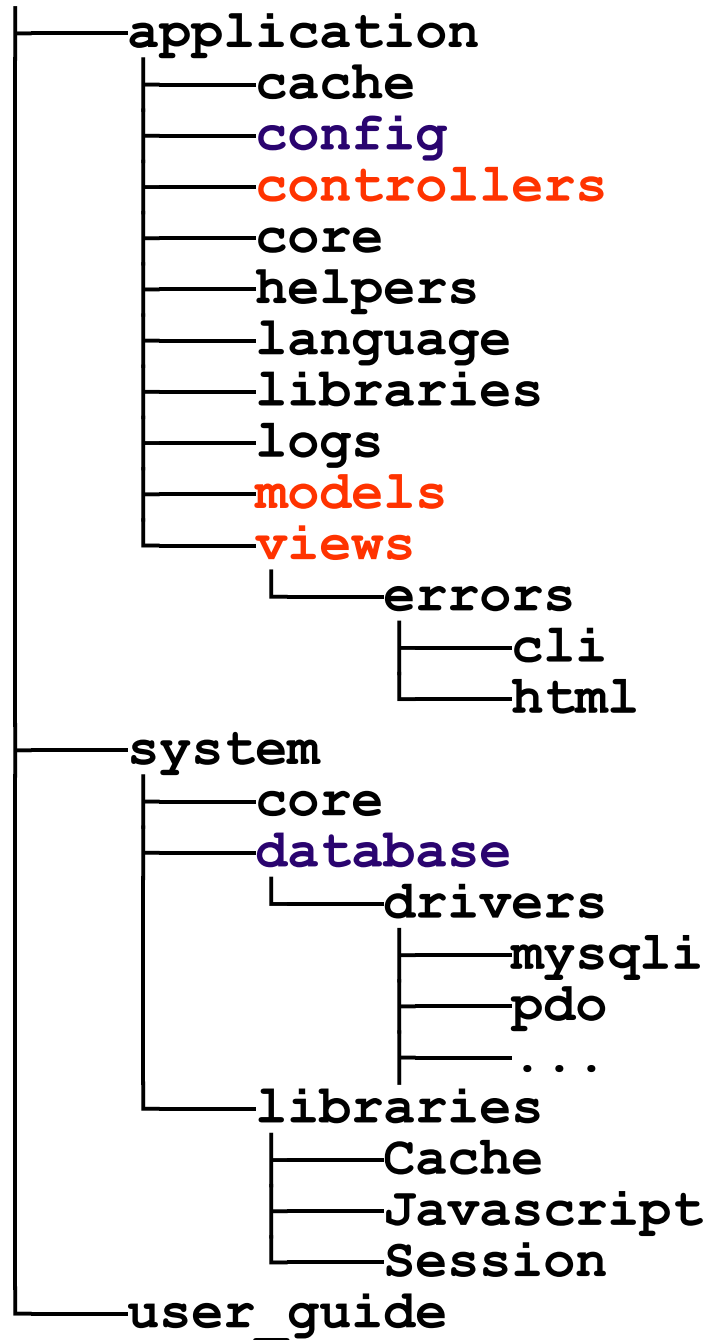
Symfony – <http://symfony.com/>

Yii – <http://www.yiiframework.com/>

Zend Framework – <http://framework.zend.com/>

altele enumerate la www.phpwact.org/php/mvc_frameworks

structura de directoare a unei aplicații Web dezvoltate cu un *framework* PHP axat pe MVC



instrumente

Managementul dependențelor
dintre biblioteci și pachete

Composer

<https://getcomposer.org/>

detalii la www.phptherightway.com/#dependency_management



maps



Packagist is the main Composer repository. It aggregates public PHP packages installable with Composer.

mediawiki/maps

JavaScript

↓ 9 703

Adds various mapping features to MediaWiki

★ 26

fwartner/maps

PHP

↓ 564

A codeigniter library that has been modified a bit so that it's laravel friendly forked by fwartner

★ 8

jasin755/maps

Tool for paste google map to the page with markers

↓ 43

★ 0

php-google-maps/php-google-maps

PHP Google Maps API for PHP 5.3+

↓ 9 991

★ 74

jstayton/google-maps-geocoder

A PHP wrapper for the Google Maps Geocoding API v3.

PHP

↓ 16 377

★ 76

alexpechkarev/google-maps

Collection of Google Maps API Web Services for Laravel

PHP

↓ 2 451

★ 9

mediawiki/semantic-maps

Adds geographic capabilities to Semantic MediaWiki

PHP

↓ 6 548

★ 17

Packagist
depozit de pachete (*repository*)
gestionate prin **Composer**
<https://packagist.org/>

instrumente

Medii pre-configurate pentru dezvoltare Web
server Web + PHP + server(e) de baze de date + utilitare

Apache + PHP + MySQL/MongoDB + Perl/Python + ...

AMPPS – <http://www.ampps.com/>

XAMPP – <http://www.apachefriends.org/>

Nginx + PHP + MariaDB + Redis + unelte de administrare

WTServer – <http://wtserver.wtriple.com/>

The screenshot displays the Cloud9 IDE interface. At the top, a status bar indicates 'Saved Version 10 - Aug 14, 2015 14:31:00'. Below this, a file explorer shows 'index.php' open. The editor contains the following PHP code:

```
1 <?php
2 for ($contor = 1; $contor <= 6; $contor++) {
3     echo '<h ' . $contor . '>Mesajul #' . $contor . '</h' . $contor . '>';
4 }
5 ?>
```

To the right, a preview window shows the rendered output of the script, displaying six messages: 'Mesajul #1' through 'Mesajul #6'. The bottom of the IDE shows a terminal with the command 'php/index.php' and the output 'Starting Apache httpd, serving https://demo-project-busaco.c9.io/php/index.php. Started apache2'.

medii de dezvoltare (și pentru) PHP „în nori”
Cloud9, Codenvy, Koding, Nitrous,...

instrumente

Editarea și execuția *online* a programelor PHP

Ideone – <http://ideone.com/>

PhpFiddle – <http://phpfiddle.org/>

a se experimenta și SQL Fiddle – <http://sqlfiddle.com/>

extensii

Hack (Facebook, din 2014)

limbaj de programare pentru HHVM, extinzând PHP

scop: creșterea productivității dezvoltatorului Web

facilități: tipuri de date explicite (*type annotations*),
generics, expresii λ , colecții (Vector, Map, Set, Pair),
tuple, programare asincronă (async) și altele

<http://hacklang.org/>

studiu de caz: Wikipedia



WIKIPEDIA
The Free Encyclopedia

Article **Talk**

Read **Edit** View history

Search



World Wide Web

From Wikipedia, the free encyclopedia

"WWW" and "The web" redirect here. For other uses of WWW, see [WWW \(disambiguation\)](#). For other uses of web, see [Web \(disambiguation\)](#).

The **World Wide Web** (**WWW**, **W3**) is an [information system](#) of interlinked [hypertext](#) documents that are accessed via the [Internet](#).^[1] It has also commonly become known simply as *the Web*. Individual document pages on the World Wide Web are called [web pages](#) and are accessed with a software application running on the user's computer, commonly called a [web browser](#). Web pages may contain text, [images](#), videos, and other [multimedia](#) components, as well as [web navigation](#) features consisting of [hyperlinks](#).

[Tim Berners-Lee](#), a [British](#) computer scientist and former [CERN](#) employee,^[2] is considered the inventor of the Web. On 12 March 1989,^[3] Berners-Lee wrote a proposal for what would eventually become the World Wide Web.^[4] The 1989 proposal was meant for a more effective CERN communication system but Berners-Lee eventually realised the concept could be implemented throughout the world.^[5] Berners-Lee and [Belgian](#) computer scientist [Robert Cailliau](#) proposed in 1990 to use hypertext "to link and access information of various kinds as a web of nodes in which the user can browse at will",^[6] and Berners-Lee finished the first website in December of that year.^[7] The first test was completed around 20 December 1990 and Berners-Lee reported about the project on the newsgroup *alt.hypertext* on 7 August 1991.^[8]

Contents [\[hide\]](#)

- 1 History
- 2 Function
 - 2.1 Linking

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikimedia Shop](#)

Interaction

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

studiu de caz: wikipedia

Scop: oferirea de conținut deschis
via o suită de aplicații Web colaborative – *wiki-uri*

Wikipedia Foundation

menține și Wiktionary, Wikinews, Wikibooks, Wikiquote,
Wikisource, Wikiversity, Wikispecies,
Wikimedia Commons, Wikidata, Wikivoyage

http://en.wikipedia.org/wiki/Wikimedia_Foundation

studiu de caz: wikipedia

MediaWiki (sistemul *wiki* utilizat pentru toate serviciile)

PHP (platforma pe care rulează MediaWiki via **HHVM**)

MySQL / MariaDB (soluția principală de stocare)

ImageMagick, DjVu, TeX, rsvg, ploticus etc.

(pentru procesare de conținuturi grafice în MediaWiki)

nginx (server Web)

Linux Ubuntu (platforma de exploatare)

se oferă inclusiv un API destinat dezvoltatorilor Web:

www.mediawiki.org/wiki/API:Main_page

studiu de caz: wikipedia

asigurarea performanței:

Squid și Varnish (*proxy & caching* pentru conținut HTML)

Memcached (*caching* interogări asupra bazelor de date)

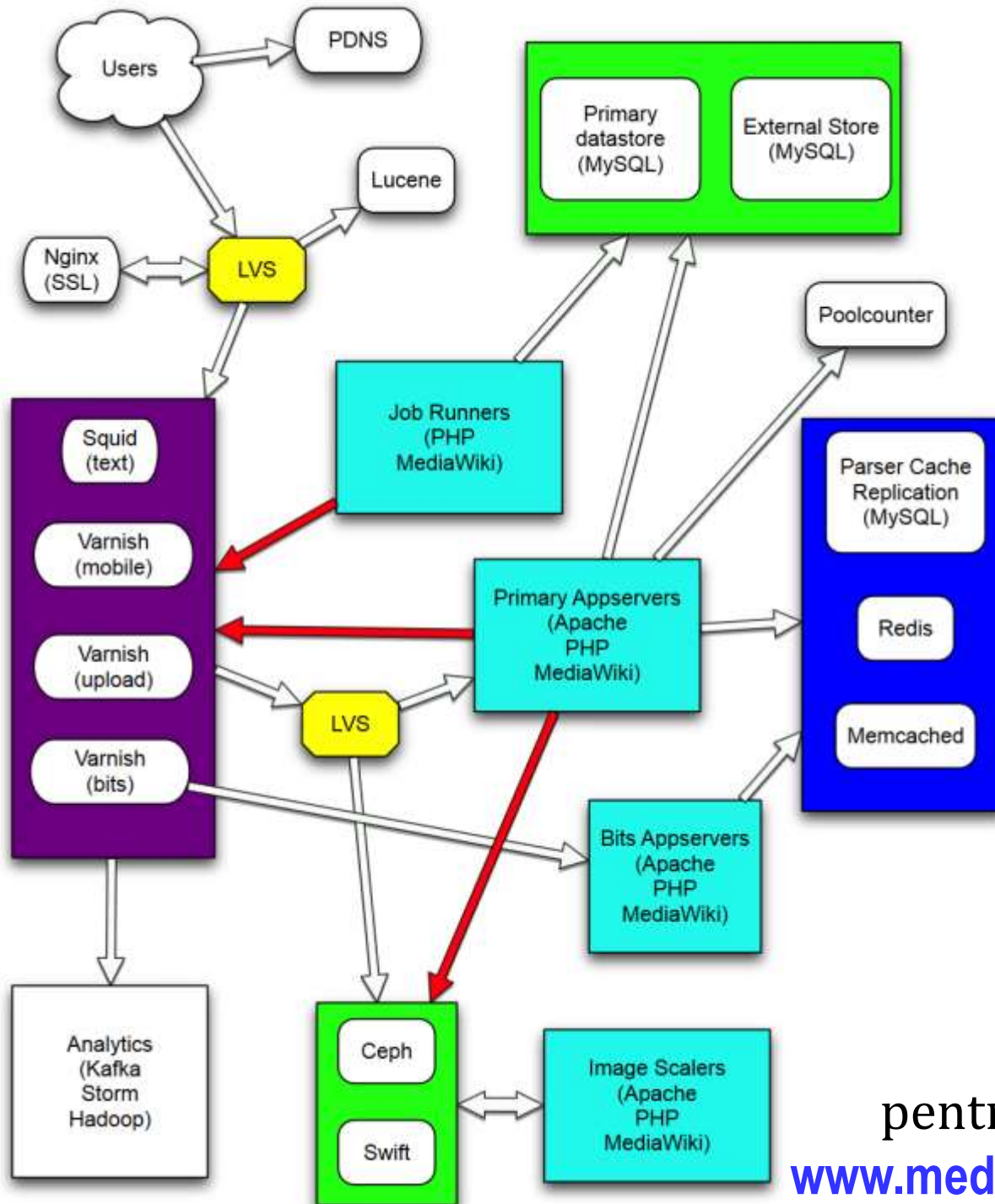
Apache Lucene (indexare textuală, facilitând căutarea)

Ceph și Swift (soluții de stocare redundantă distribuită)

Linux Virtual Server – LVS (*load balancing*)

PowerDNS (soluție C++ pentru DNS)

http://meta.wikimedia.org/wiki/Wikimedia_servers



pentru detalii, a se consulta

www.mediawiki.org/wiki/Presentations

rezumat

privire generală asupra PHP

() () () ()
() () () ()
() () () ()

caracterizare, facilități, instrumente

NODE.JS MODULES

POPULAR **INTERESTING** NEW

TAGS

#1 **npm**

98.4% A package manager for node

#2 **express**

98.2% Sinatra inspired web development framework

#3 **mocha**

97.3% simple, flexible, fun test framework

#4 **jade**

97.2% Jade template engine

jade: Jade template engine

#5 **connect**

96.9% High performance middleware framework

#6 **socket.io**

95.3% Real-time apps made cross-browser & easy with a WebSocket-like...

#7 **mongodb**

95.3% A node.js driver for MongoDB

#8 **stylus**

95.0% Robust, expressive, and feature-rich CSS superset

Find tag...

all javascript client api simple
browser server framework web
module parser http nodejs
implementation test css json file
express data cli tool engine
testing coffeescript object
database html based middleware
wrapper command connect
template easy language files
asynchronous async build code
fast stream ender line mongodb
utility generator compiler rest
system applications jquery
templates dom support bdd
lightweight objects tools bindings

episodul viitor:

dezvoltarea de aplicatii Web cu Node.js