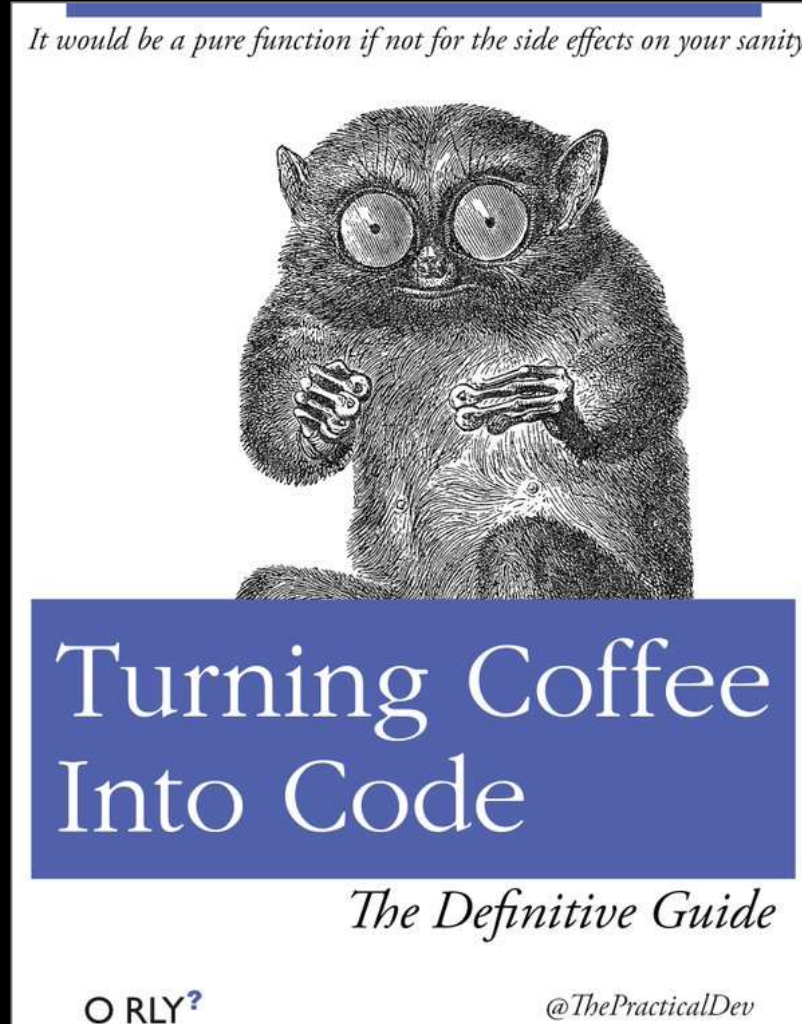


Tehnologii Web



programare Web (III)
dezvoltarea aplicațiilor Web cu **PHP**

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

Leonardo da Vinci

Cum folosim un server de aplicații
pentru a dezvolta o aplicație Web?

server de aplicații web

Scop:
eficientizarea proceselor de dezvoltare
a aplicațiilor Web de anvergură

server de aplicații web

Integrat în unul/mai multe servere Web

de asemenea, poate oferi propriul server Web
sau mediu de execuție

server de aplicații web

Poate încuraja sau impune o viziune arhitecturală privind dezvoltarea de aplicații Web

situație tipică:
MVC ori variații



(re)vezi prezentarea
anterioară

server de aplicații web

Simplifică maniera de invocare
de programe (*script-uri*) ale unei aplicații Web

- ▶ generarea de conținut dinamic pe partea de server

server de aplicații web

Aspecte de interes:

limbaj(e) de programare

API-ul de bază

stocare persistentă a modelelor de date

interacțiune Web

cookie-uri și sesiuni

medii de dezvoltare + cadre de lucru, componente,...

caracteristici particulare

server de aplicații web

Limbaj(e) de programare

oferind suport pentru una sau mai multe **paradigme**:

imperativă

obiectuală

funcțională

reactivă

concurentă

paralelă

...

server de aplicații web

Limbaj(e) de programare

statice – exemple: C#, Java, Rust

versus

dinamice – *e.g.*, JavaScript, PHP, Python, Ruby

server de aplicații web

Limbaj(e) de programare

procesate prin **interpretare** și/sau **compilare**

uzual, se preferă generarea de **cod intermediar**:
IL (*Intermediate Language*) – C#, Java, Scala,...

server de aplicații web

API de bază

contribuie la „puterea” limbajului și
a serverului de aplicații
(via funcții/clase predefinite)

server de aplicații web

API de bază

contribuie la „puterea” limbajului și
a serverului de aplicații
(via funcții/clase predefinite)

securitate, consistență,
acces la resursele mediului de operare/rulare,
asigurarea independenței de platformă etc.

server de aplicații web

Stocare persistentă

în baze de date relaționale – folosind **SQL**

server de aplicații web

Stocare persistentă

în baze de date relaționale – folosind **SQL**

exemple:

ADO.NET pentru ASP.NET

Java – **JDBC** (*Java DataBase Connectivity*)

PHP – funcții/module predefinite, plus biblioteci
incorporate (**SQLite** + **mysqli**) sau diverse extensii

server de aplicații web

Stocare persistentă

în baze de date relaționale – folosind **SQL**

ORM (*Object-Relational Mapping*)

Go – *framework*-ul **Xorm**

Java – specificația **JPA** (*Java Persistence API*)

+ implementări: **EclipseLink**, **Hibernate**, **OpenJPA**,...

Node.js – biblioteca **Sequelize**

PHP – *framework*-uri: **Doctrine**, **Propel** etc.

server de aplicații web

Stocare persistentă

în baze de date relaționale – folosind **SQL**

Active Record utilizat în cadrul ORM

exemple:

active_record (modul Node.js), **Castle Project** (.NET),
DBIx::Class (Perl), **Orator** (Python),
Play Framework (Java, Scala), **Rails** (Ruby)

server de aplicații web

Stocare persistentă

pe baza modelelor arborescente: XML

date (semi)structurate

transformări în alte formate: XPath, XSLT

procesări: DOM, SAX, SimpleXML etc.

validări de date: DTD, XML Schema, RELAX,...

interogări: XQuery

cursurile
viitoare

server de aplicații web

Stocare persistentă

recurgând la alte paradigme non-relaționale
(bazate pe grafuri și/sau cheie—valoare)

distribuite la nivel de Internet, scalabile – NoSQL

github.com/ericleung/awesome-nosql

nosql.mypopescu.com/kb/nosql

exemplificări:

Cassandra, CouchDB, Hadoop, MarkLogic, MongoDB, Neo4j etc.

server de aplicații web

Interacțiune Web

facilitată de controale specificate
în cadrul codului-sursă rulat la nivel de server

se pot emula câmpuri din formularele HTML și/sau
oferi noi controale interactive – *e.g.*, calendar, *slideshow*,...

- ▶ generare de cod procesabil la nivel de client (*front-end*)
HTML (+JavaScript) în funcție de navigatorul Web

server de aplicații web

Interacțiune Web

exemplificări:

ASP.NET (**<asp:control>** – *e.g.*, FileUpload, ListBox, Table,...)

framework-ul **PRADO** (PHP)

formidable, form-data, forms – module Node.js

platforma Java: **JSF** (*JavaServer Faces*)

server de aplicații web

Interacțiune Web

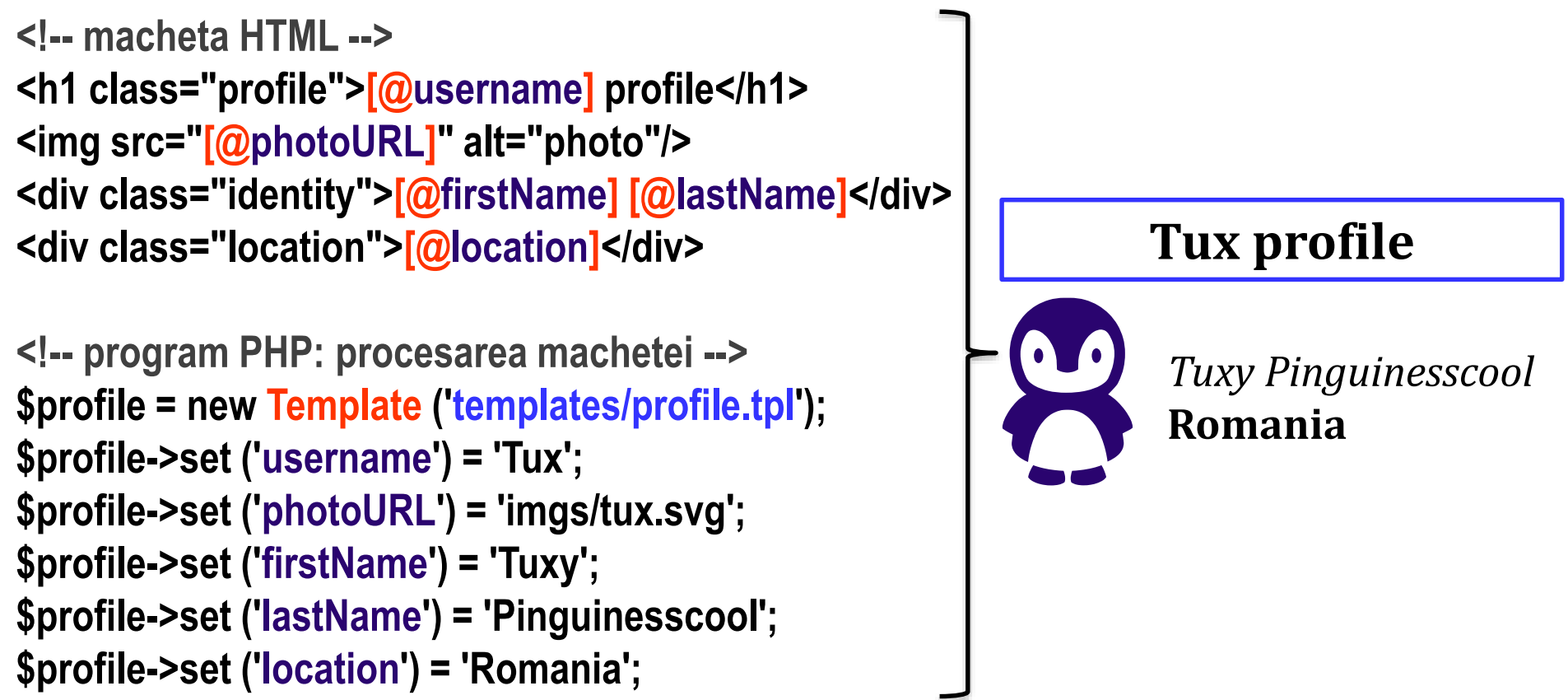
încurajarea folosirii de machete de vizualizare (*templates*)
pe baza unui procesor specific – ***Web template system***

server de aplicații web

Interacțiune Web

Web template system

utilizând specificații de prezentare a conținutului (*Web template*), datele persistente (*e.g.*, preluate dintr-o bază de date) sunt folosite de un procesor (*template engine*) pentru a genera documente HTML ori alte formate



exemplu:

specificația de prezentare a conținutului

(*HTML template* – **.tpl**) include nume de variabile – aici,
de forma **[@variabilă]** – ce vor fi înlocuite cu valorile efective
preluate din program în urma rulării componentei
de procesare a machetelor de vizualizare

server de aplicații web

Interacțiune Web

Web template system

la nivel de server

Apache FreeMarker (Java), Blade (PHP), Haml (Ruby),
Mustache (C++, JS, PHP, Python, Scala,...), Razor (.NET),
Smarty (PHP), Tonic (PHP), Velocity (Java), XSLT (XML)

server de aplicații web

Interacțiune Web

Web template system

la nivel de client

disponibile pentru JavaScript:

Dust.js, EJS, Handlebars, Mustache.js,...

github.com/sorrycc/awesome-javascript#templating-engines

server de aplicații web

Interacțiune Web

transfer asincron de date via suita de tehnologii Ajax



vezi cursurile
viitoare

eventual, via *framework*-uri/module/clase adiționale

server de aplicații web

Suport acordat ingineriei software

stimularea folosirii unor **șabloane de proiectare**:

Container

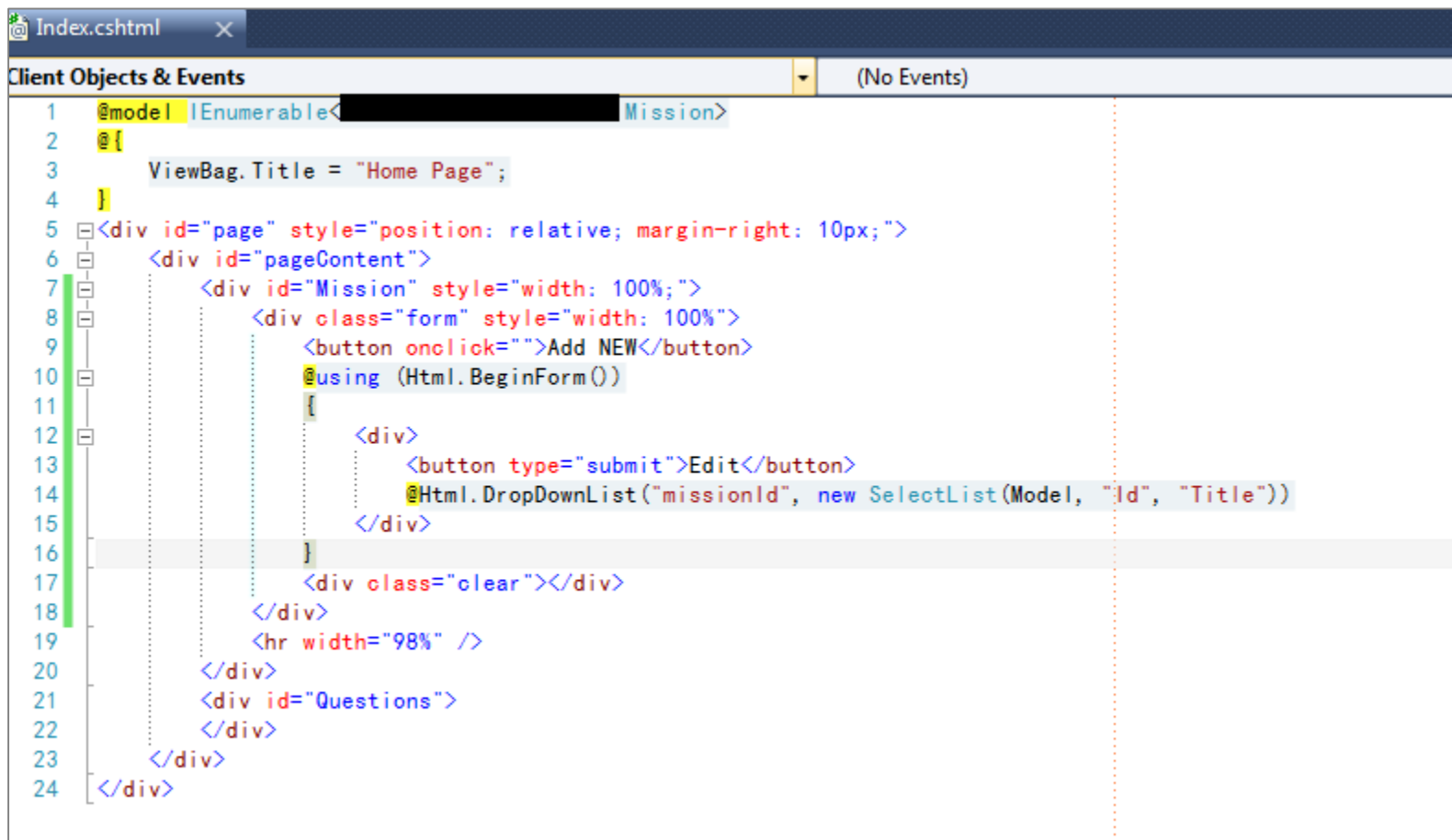
MV*

Proxy

Configuration Parameters

Invocation Context

...



```
1 @model IEnumerable<Mission>
2 @
3 ViewBag.Title = "Home Page";
4
5 <div id="page" style="position: relative; margin-right: 10px;">
6     <div id="pageContent">
7         <div id="Mission" style="width: 100%;">
8             <div class="form" style="width: 100%">
9                 <button onclick="">Add NEW</button>
10                 @using (Html.BeginForm())
11                 {
12                     <div>
13                         <button type="submit">Edit</button>
14                         @Html.DropDownList("missionId", new SelectList(Model, "Id", "Title"))
15                     </div>
16                 }
17                 <div class="clear"></div>
18             </div>
19             <hr width="98%" />
20         </div>
21         <div id="Questions">
22         </div>
23     </div>
24 </div>
```

exemplificare: **anti-șablonul** *Spaghetti Code*
uzual, specific aplicațiilor Web care „amestecă”
partea de procesare cu maniera de prezentare (*view*)
și mecanismul de acces la modelul datelor
sourcemaking.com/antipatterns/spaghetti-code

server de aplicații web

Suport acordat ingineriei software

încurajează adoptarea unei/unor arhitecturi:
stratificate (*layered*)
conduse de evenimente (*event-driven*)
extensibile (*microkernel / plug-in*)
folosind microservicii (*microservices*)
„în nori” (*space-based, cloud*)

server de aplicații web

Exemple – inclusiv *framework*-uri

C++ – CppCMS, Silicon, TreeFrog, Wt

C# *et al.* (platforma .NET) – ASP.NET

D – Dweb, vibe.d

Dart – Aqueduct, shelf, Stream,...

Erlang / Elixir – Chicago Boss, Phoenix, N2O, Sugar

Go – Beego, Buffalo, Gin, Gorilla, Revel

Haskell – Snap, Spock, Yesod etc.

server de aplicații web

Exemple – inclusiv *framework*-uri

Java (*Jakarta Enterprise Edition* – ex-J2EE) – GlassFish, JBoss, Tomcat, Payara + Apache Struts, Apache Wicket, Grails, Spring, Vaadin

JavaScript (ECMAScript) – Node.js + Derby, Express, Locomotive, Meteor,...

Perl – Catalyst, CGI::Application, Mojolicious

PHP – PHP

server de aplicații web

Exemple – inclusiv *framework*-uri

Python – Gunicorn, Tornado, uWSGI, Waitress etc.

(pe baza specificației WSGI – *Python Web Server Gateway Interface*)

+ Django, TurboGears, web2py

Ruby – Passenger, Puma, Thin, Unicorn,...

(implementează specificația Rack)

+ Ruby on Rails, Sinatra

Rust – Gotham, Iron, Rocket, Rouille etc.

Scheme – Artanis

server de aplicații web

Exemple – inclusiv *framework*-uri

abordare multi-limbaj:

Vert.x

(disponibil pentru Java, JS, Groovy, Ruby, Ceylon, Scala,...)

server de aplicații web

Integrat într-o *stivă de tehnologii (software stack)*

suită software (servere, instrumente, utilitare,...)
oferind suport pentru dezvoltarea de aplicații Web

disponibilă – uzual, în regim *open source* –
pentru o anumită platformă
(sistem de operare, server Web, server de baze de date,
server de aplicații, limbaj de programare)

server de aplicații web

Integrat într-o stivă de tehnologii (*software stack*)










LAMP

(Linux, Apache HTTP Server, MariaDB/MongoDB, Perl/PHP/Python)

alternative:

FAMP (FreeBSD), MAMP (macOS),
WAMP (Windows), XAMP (multi-platformă)

www.apachefriends.org

	Apache	2.4.45
	MariaDB	10.1.22 10.1.23 10.2.6
	Memcached	1.4.5
	MongoDB	3.2.13
	MySQL	5.7.18
	Nginx	1.12.0 1.13.0 1.13.1
	Php	5.6.29 5.6.30 5.6.33 7.0.18 7.0.19 7.0.20 7.0.27 7.1.4 7.1.5 7.1.6 7.1.14 7.2.2
	PostgreSQL	9.6.3
	Redis	3.0.504

www.wamp.net – instrumente dedicate dezvoltatorilor Web

server de aplicații web

Integrat într-o stivă de tehnologii (*software stack*)

MEAN – *full stack JavaScript*
(MongoDB, Express.js, Angular, Node.js)

mean.io

server de aplicații web

Integrat într-o stivă de tehnologii (*software stack*)

abordări complementare:

LAPP (Linux, Apache, PostgreSQL, Perl/PHP/Python)

LEMP (Linux, Nginx, MySQL/MariaDB, Perl/PHP/Python)

LLMP (Linux, Lighttpd, MySQL/MariaDB, Perl/PHP/Python)

LYCE (Linux, Yaws, CouchDB, Erlang)

LYME (Linux, Yaws, Mnesia, Erlang)

Informații esențiale despre PHP?

php

Istoric

Caracteristici importante

Limbajul de programare PHP

paradigme: procedurală, obiectuală, funcțională

PHP ca platformă de dezvoltare Web

interacțiune, acces la baze de date, *framework*-uri,
biblioteci și instrumente, studii de caz concrete

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), PHP 7.1 (2016), PHP 7.2 (2017)
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)

poate fi folosit și ca limbaj de uz general

php: caracterizare

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

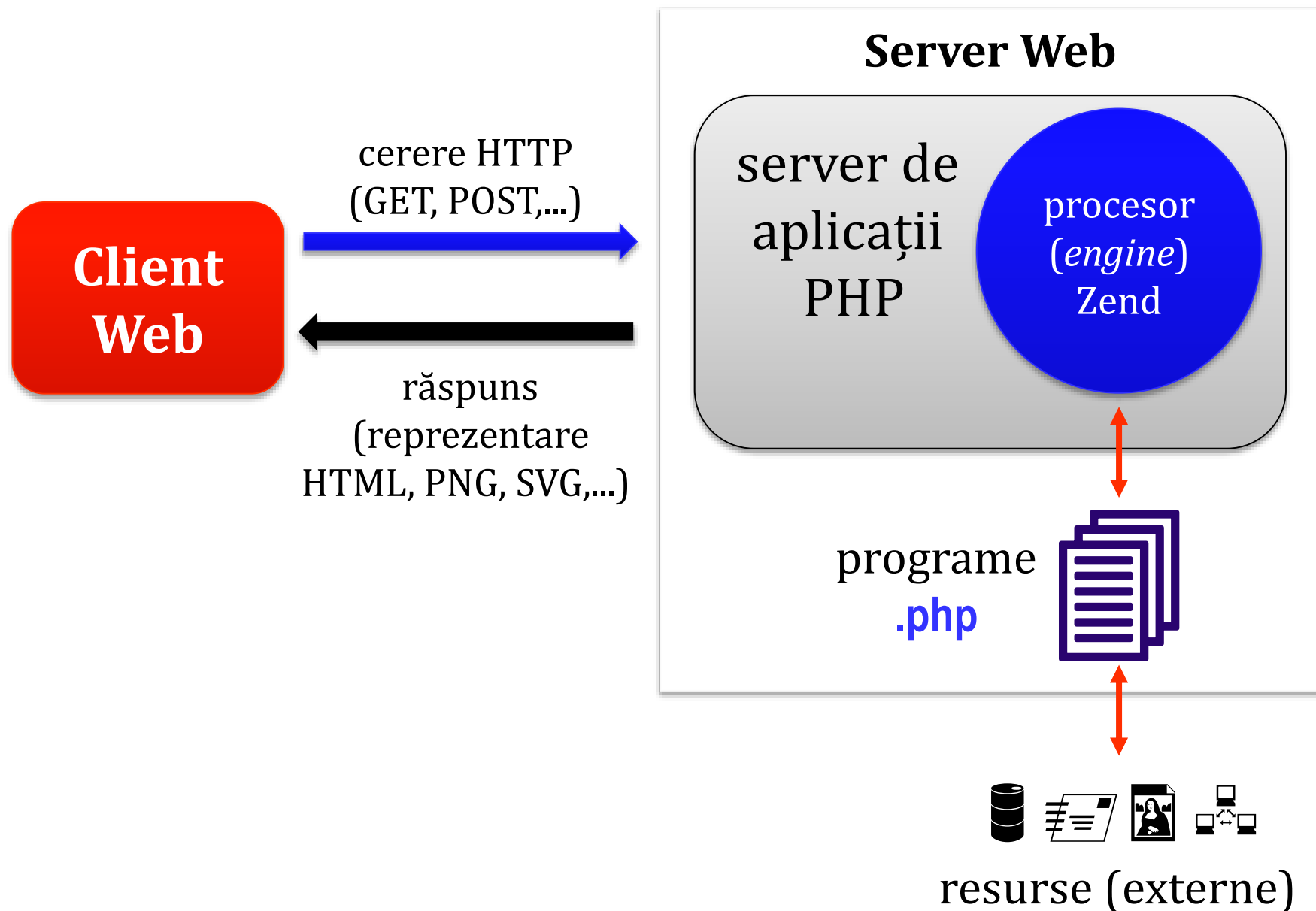
uzual, fișierele ce conțin cod-sursă PHP au extensia **.php**

php: caracterizare

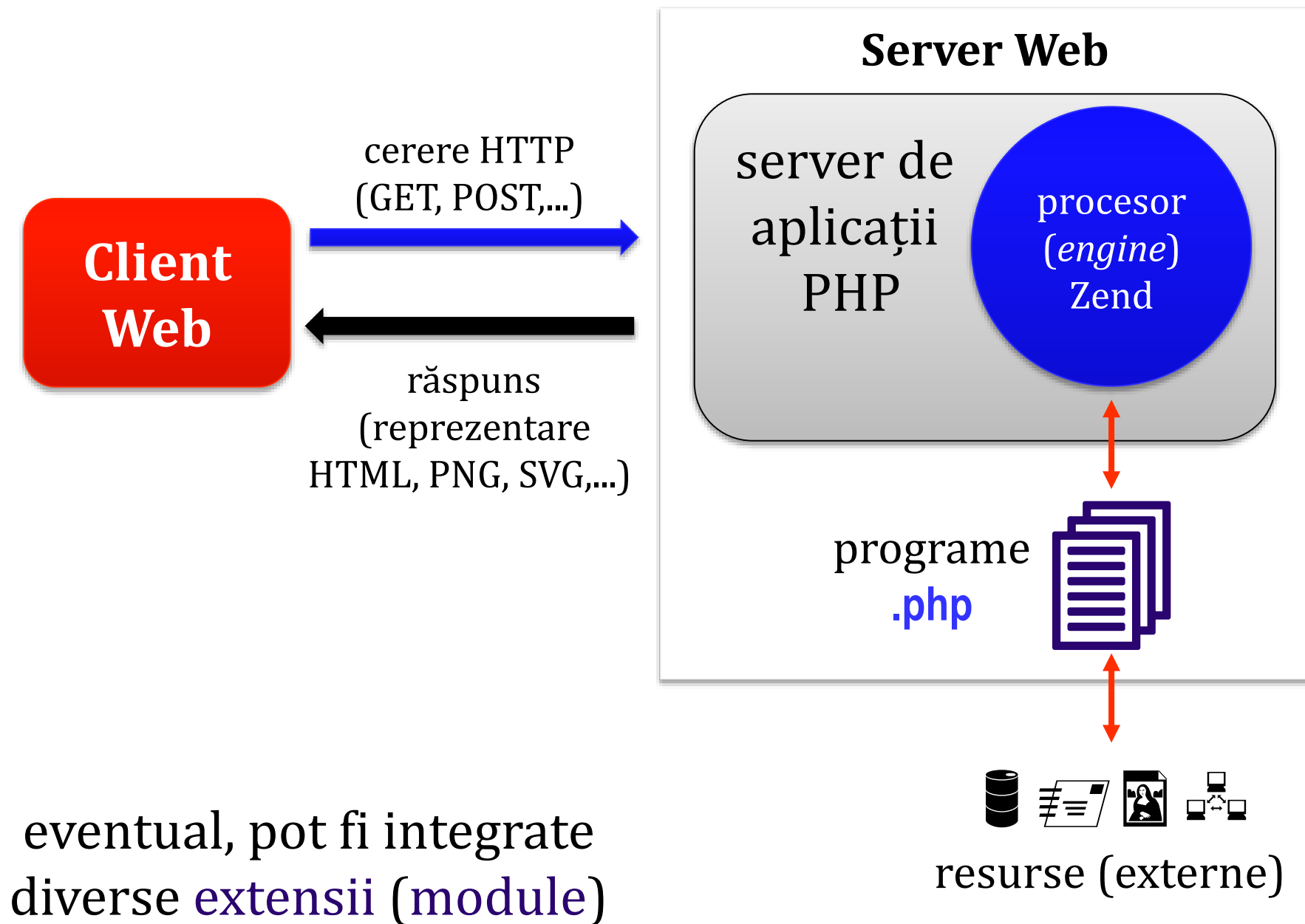
Disponibil gratuit – *open source* – pentru diverse platforme (Linux, Windows, macOS, UNIX) și servere Web: Apache, IIS, NGINX,...

www.php.net
www.zend.com

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



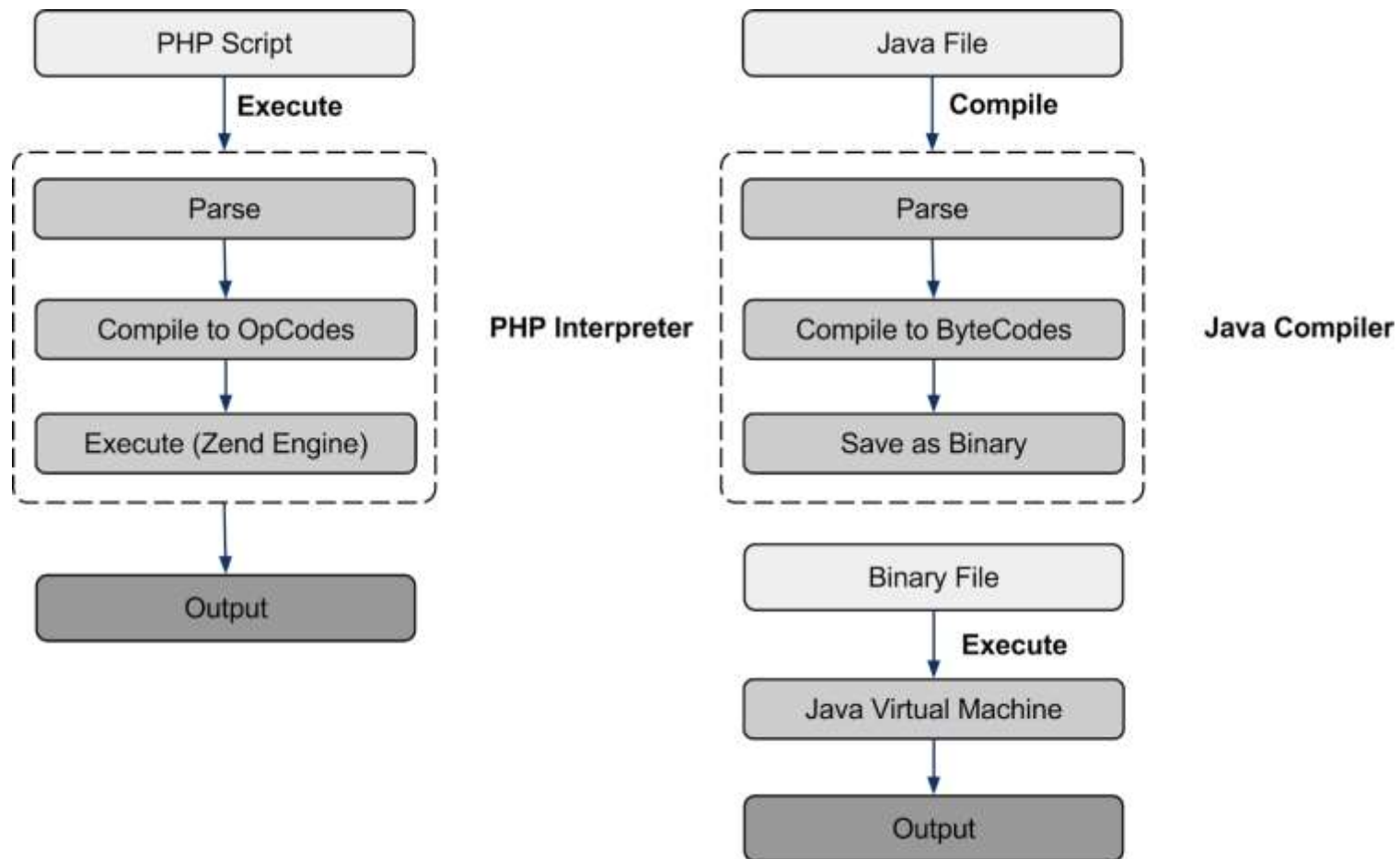
Maniera de funcționare a procesorului (*engine*-ului) PHP



php: caracterizare

Programul PHP e interpretat de Zend Engine 2
care generează instrucțiuni interne – *opcodes*

www.php.net/manual/en/internals2.opcodes.php
www.phpinternalsbook.com/



fazele importante ale interpretării programelor PHP
vs. compilarea codului Java

codul PHP e interpretat de fiecare dată când trebuie rulat

<?php

```

class Greeting {
    public function sayHello ($to)
    {
        echo "Hello $to";
    }
}

```

```

$greeter = new Greeting ();
$greeter->sayHello ("World");
?>

```

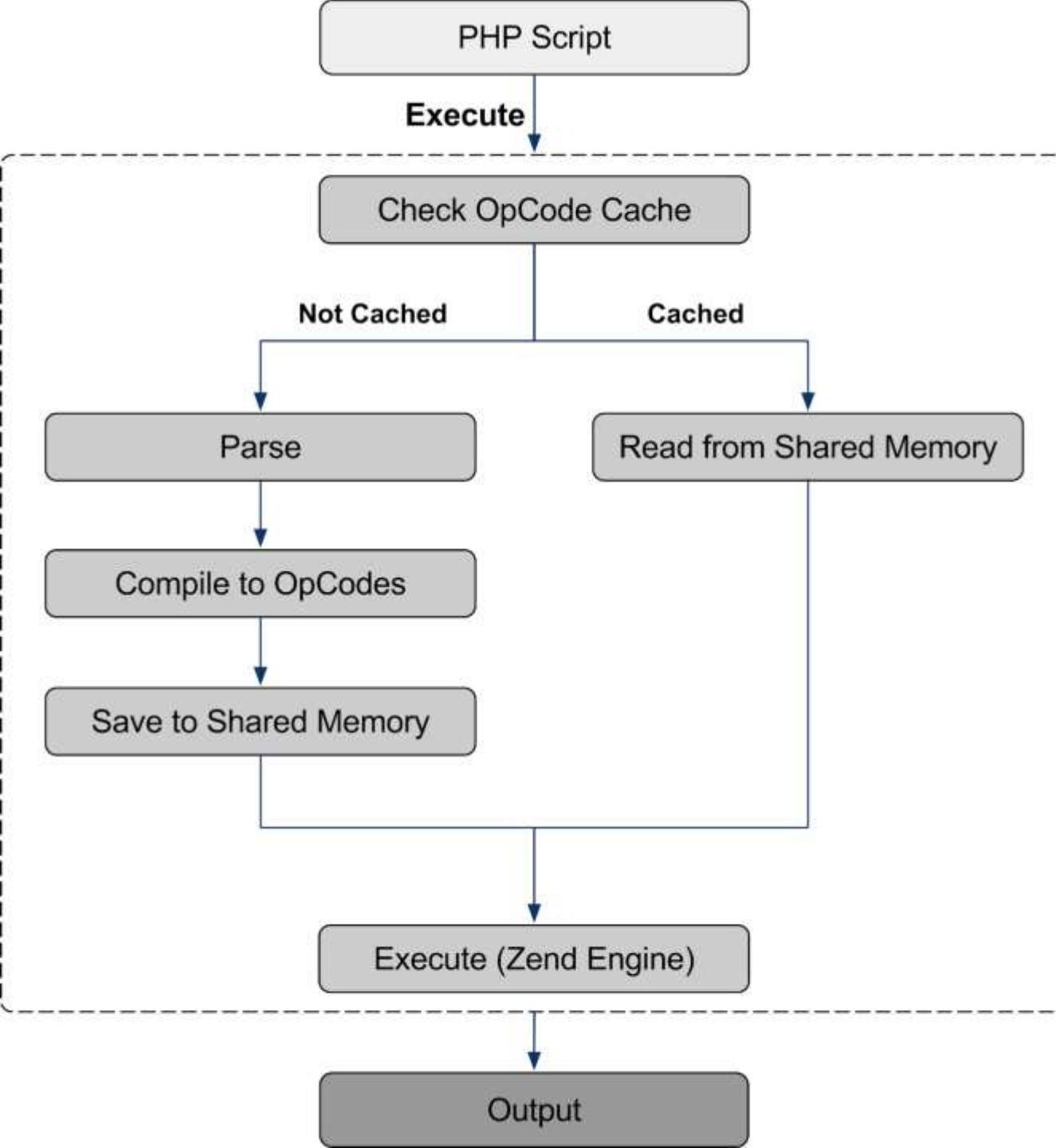
convertirea codului PHP
în *opcodes*
tinyurl.com/zn6c53x

line	#	*	op	fetch	ext	return	operands
3	0	>	EXT_NOP				
	1		RECV			!0	
5	2		EXT_STMT				
	3		ADD_STRING			~0	'Hello+'
	4		ADD_VAR			~0	~0, !0
	5		ECHO				~0
6	6		EXT_STMT				
	7	>	RETURN				null



opcodes

avansat



pentru eficiență,
opcode-urile
sunt stocate
într-o memorie
partajată

precision = 14 ; controlul preciziei valorilor float – detalii la php.net/precision
safe_mode = Off ; controlul procesării – de studiat 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-8859-2');

// 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)

cod sursă PHP ► *opcodes* ► cod mașină (*e.g.*, x86-64)

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

DBAL (*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 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

conținut grafic în diverse formate

fișiere în format JSON

documente XML – creare, procesare, validare etc.

cărți de credit

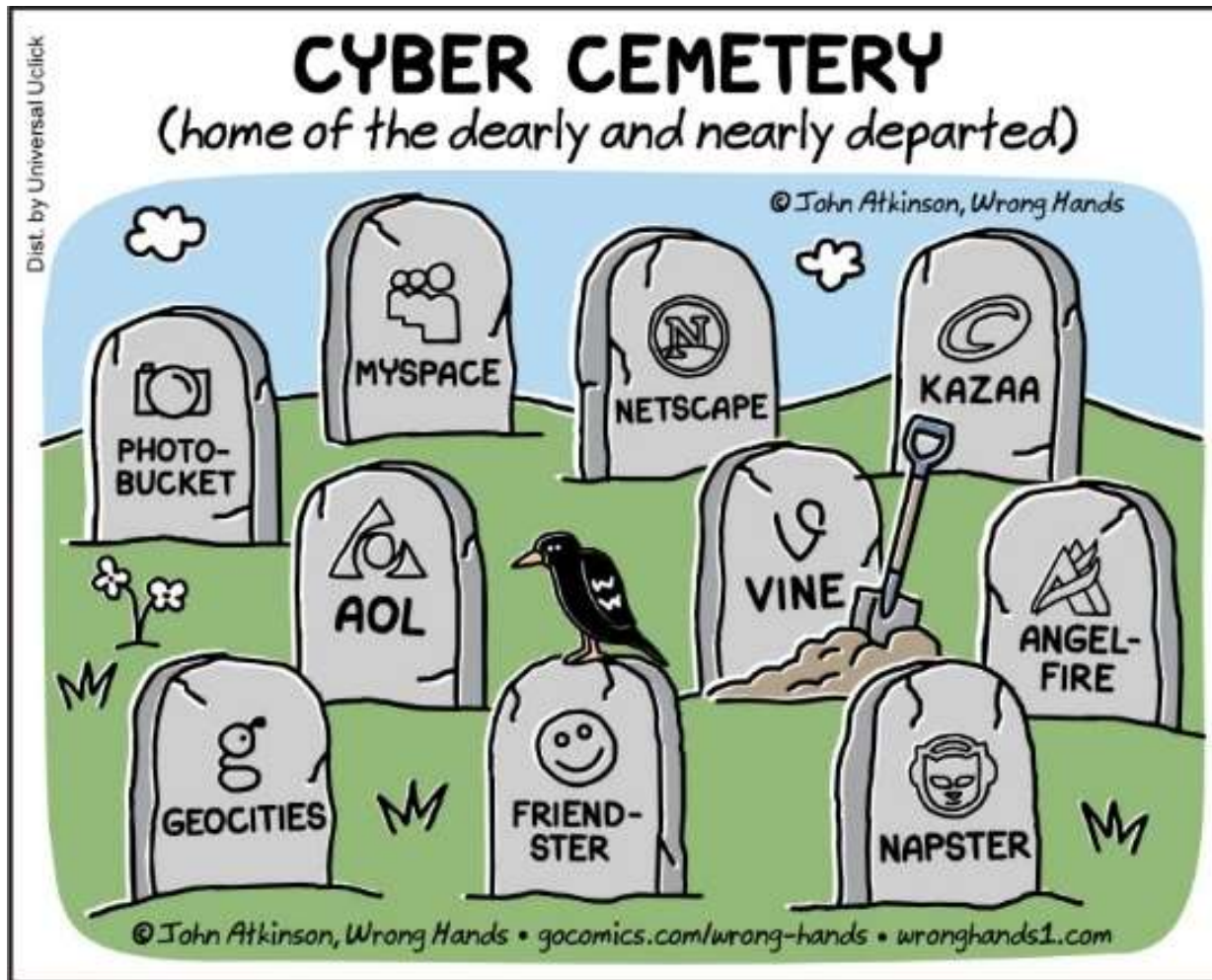
...

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

(în loc de) pauză



wronghands1.files.wordpress.com/2018/09/cyber-cemetery.jpg

PHP ca limbaj de programare procedurală

php: caracterizare

Programare procedurală (*procedural programming*)

paradigmă bazată pe apeluri de proceduri (rutine, funcții) conținând o serie de pași pentru realizarea calculelor

limbajele procedurale sunt imperative, utilizând instrucțiuni (comenzi) ce modifică starea programului

exemple: FORTRAN (1954), ALGOL (1958),
BASIC (1964), Pascal (1970), C (1972), Ada (1978)

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
(doar PHP 7 are suport nativ pentru Unicode)

php: tipuri de date – scalare

string

șiruri de caractere ASCII
(doar PHP 7 are 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)

php: tipuri de date – compuse

array

nu există o distincție clară
între tablouri indexate și cele asociative

un tablou poate reprezenta diverse structuri de date:
listă (vector), tablou asociativ – *hash* (implementarea
unei asocieri de valori – *mapping*), dicționar, colecție,
stivă, coadă,...

php: tipuri de date – compuse

array

// un tablou indexat (vector de valori)

```
$cadouri = array ("ceas", "prăjitură", "colier", "topor");
```

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

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

// sintaxa simplificată – începând cu 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

exemplificări: bzip2, curl, ftp, gd, mysql link, mysql result, pdf document, printer, stream, socket, xml, zlib

o resursă e creată de funcții specifice

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

php: tipuri de date – speciale

resource

semnifică o referință la o resursă externă

funcții predefinite:

`is_resource()`

`get_resource_type()`

detalii la 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()`

`unset()`

php: variabile

Variabilele au nume compuse din litere, cifre și caractere _ prefixate de simbolul \$

pot stoca valori – aparținând unui tip de date – sau referințe

www.php.net/manual/en/language.variables.php

php: variabile

Variabile create „din zbor”

tipul de date e determinat pe baza contextului

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

<code>\$ani = 21;</code>	<code>/* o variabilă de tip Integer */</code>
<code>\$conectat = TRUE;</code>	<code># una de tip Boolean</code>
<code>\$prefer["culoare"] = "gri";</code>	<code>// un tablou asociativ</code>

php: variabile

Funcții predefinite utile:

`var_dump()`

`settype()`

`is_bool()`, `is_int()`, `is_float()`, `is_array()`, `is_string()`

`is_scalar()`, `is_numeric()`

...și altele

php: variabile

Vizibilitatea variabilelor (*scope*)

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

php.net/manual/en/language.variables.scope.php

```
$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

`$_SERVER []`

`$_GET [] $_POST [] $_FILES [] $_REQUEST []`

`$_SESSION []`

`$php_errormsg`

`$argc $argv`

...

php: variabile predefinite

specifice
serverului Web

... `$_SERVER []`

`$_GET []` `$_POST []` `$_FILES []` `$_REQUEST []`

date despre
sesiunea Web

... `$_SESSION []`

cereri HTTP
ale clientului

`$php_errormsg`

mesaj de eroare
raportat

argumente în
linia de comandă

... `$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

DIRECTORY_SEPARATOR

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

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

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 ("Numele nu a fost furnizat...");  
} else {  
    echo ("Bine ai venit, " . $nume . "!\n");  
}
```


php: exemplu

```
<?php
```

```
// umplem un tablou cu valori de la 1 la 10
```

```
for ($index = 1; $index <= 10; $index++) {  
    $values[$index] = $index;  
}
```

```
// realizăm suma valorilor
```

```
$sum = 0;
```

```
foreach ($values as $item)
```

```
    $sum += $item;
```

```
/* afișăm suma obținută la ieșirea standard  
   pentru a fi trimisă clientului Web */
```

```
echo("<p>Sum of first 10 numbers is <strong>" .  
    $sum . "</strong></p>");
```

```
?>
```

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

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

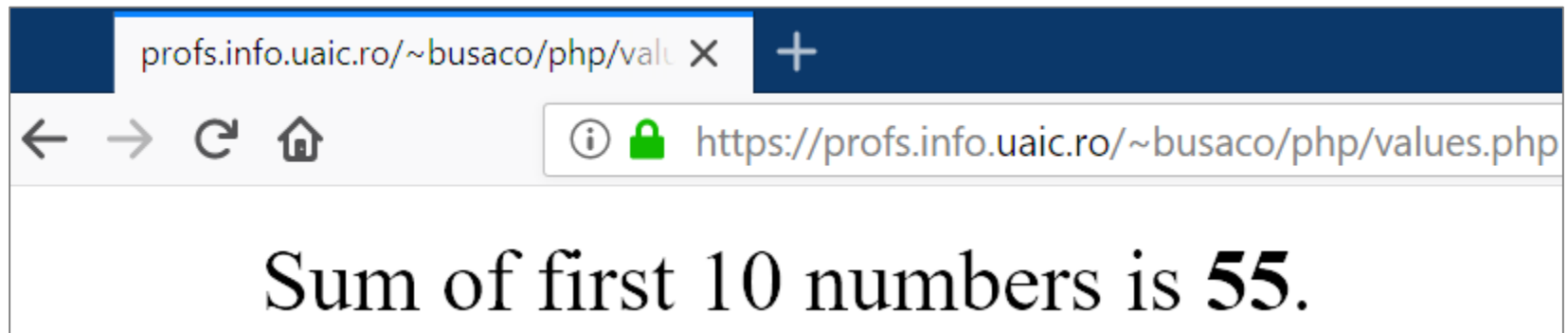
php values.php

<p>Sum of first 10 numbers is 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*

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

```
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 ...

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

pentru PHP 7+, se poate preciza și tipul de date
pentru fiecare argument, plus 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
criptografice
pentru accesarea resurselor XML, PDF, JPEG,...
specifice sistemului de operare
generale

detalii la 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()

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:

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: www.pcre.org

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

REGULAR EXPRESSION

1 match, 22 steps (-0ms)

/ (\w+\W)+(\d{2,4})\$/igu

TEST STRING

SWITCH TO UNIT TESTS ▶

WWW inception 1989

EXPLANATION

- ▼ / (\w+\W)+(\d{2,4})\$/igu
 - ▼ **1st Capturing Group** (\w+\W)+
 - + Quantifier** — Matches between **one** and **unlimited** times, as many times as possible, giving back as needed (greedy)
 - A repeated capturing group will only capture the last iteration. Put a capturing group around the repeated group to capture all iterations or use a non-capturing group instead if you're not interested in the data.

MATCH INFORMATION

Match 1

Full match	0-18	WWW inception 1989
Group 1.	4-14	inception
Group 2.	14-18	1989

QUICK REFERENCE

Search reference

- All Tokens
- ★ **Common Tokens** ✓
- General Tokens

A single character of: a, b o... [abc]
 A character except: a, b or c [^abc]
 A character in the range: a-z [a-z]
 A character not in the ran... [^a-z]

un instrument Web
de testare și depanare
a expresiilor regulate
regex101.com/#PCRE

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()`

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) { //closure: 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 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()`
`strtotime()`, `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 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_writable()`,...
`chdir()`, `mkdir()`, `rmdir()`
`disk_free_space()`, `disk_total_space()`

de studiat și 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

Suport pentru operații criptografice:

password hashing – funcțiile `password_*`()

extensii utile

CSPRNG (generare de numere pseudo-aleatoare – la PHP 7+)

Hash (funcții `hash_*`() – de tip *digest*)

OpenSSL (funcționalități ce vizează SSL/TLS)

Sodium (operații de (de)criptare avansate – cazul PHP 7.2+)

amănunte la php.net/manual/en/refs.crypto.php

php: funcții predefinite

Suport vizând conținutul grafic (raster/vectorial):

extensii preinstalate

Cairo – procesări vectoriale/raster: www.cairographics.org

EXIF – acces la meta-date JPEG

GD – prelucrări raster (GIF, JPEG, PNG): libgd.github.io

ImageMagick – procesări multi-format: www.imagemagick.org

php.net/manual/en/refs.utilspec.image.php

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

Execuția programelor din linia de comandă
PHP CLI

php.net/manual/en/features.commandline.php

php: alte facilități

PHP ca modul al unui server Web (Apache, NGINX)

e.g., pe baza modulelor **proxy_fcgi**
și **php-fpm** (*FastCGI Process Manager*) – php-fpm.org

wiki.apache.org/httpd/php

www.nginx.com/resources/wiki/start/topics/examples/phpfcgi/

php: alte facilități

PHP 5.4+ oferă un server Web incorporat

exemplu de invocare:

```
php -S localhost:8000 -t phpwebapp/
```

php.net/features.commandline.webserver

php: alte facilități

Inter-conectivitatea cu alte tehnologii/platforme

exemple: Java, JavaScript, Lua, .NET

Care-i suportul oferit de PHP
pentru programarea obiectuală?

php: caracterizare

Programare obiectuală (*object-oriented programming*)

paradigmă bazată pe conceptul de **obiect** – incluzând **date** (atribute, proprietăți) și **cod** (metode, proceduri)

uzual, obiectele interacționează între ele
și reprezintă instanțe de **clase**

exemple: Smalltalk (1972), Objective-C (1984),
C++ (1985), Python (1990), Java (1995), C# (2000)

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)

php.net/manual/en/oop5.intro.php
detalii la 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 parcurge
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 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

www.php.net/manual/en/reserved.interfaces.php

php: interfețe & clase predefinite

// exemplificare: 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: introspecție

Acces la informații vizând o clasă:

un program PHP poate obține date referitoare la clase, interfețe, funcții, metode, extensii – *reverse engineering*

ReflectionClass implements **Reflector**

php.net/manual/en/book.reflection.php

php: introspecție

```
$clasa = new ReflectionClass ('StudentDestept');  
// afișăm informații despre clasa specificată  
printf ("<p>Clasa <em>%s</em> extinde %s  
      și e declarată în fișierul <tt>%s</tt>.</p>",  
$clasa->getName (), var_export ($clasa->getParentClass (), 1),  
$clasa->getFileName ());
```

► Clasa *StudentDestept* extinde

ReflectionClass::__set_state(array('name' => 'Student',))

și e declarată în fișierul </home/profs/busaco/html/php/introspect.php>.

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

astfel, se oferă suport pentru pseudo-moștenire multiplă

aspecte formale: scg.unibe.ch/research/traits

// trăsături (comportamente) 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 {
    // folosește trăsăturile dorite
    use Colorare, Mutare, Rotire; // poate fi colorat, mutat, rotit

    public function transforma () { // o transformare specifică
    }
}
```

```
// clasa Cerc nu mai poate fi extinsă
final class Cerc extends Figura {
    // un cerc poate fi mutat și colorat
    use Mutare, Colorare;

    // declararea unei constante proprii
    const PI = 3.1415265;

    // specificarea unei metode
    public function calculeazaArie () {
    }
}
```

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)

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;
```


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']`

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*
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 ) : bool  
    public string create_sid ( void ) : string  
    public string read ( string $session_id ) : string  
    public bool  write ( string $session_id , string $session_data ) : bool  
    public bool  gc ( int $maxlifetime ) : int           // elimină vechile sesiuni  
    public bool  destroy ( string $session_id ) : bool  
    public bool  close ( void ) : bool  
}
```

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

O mai bună securitate

php: bd – extensia **mysqli**

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**:

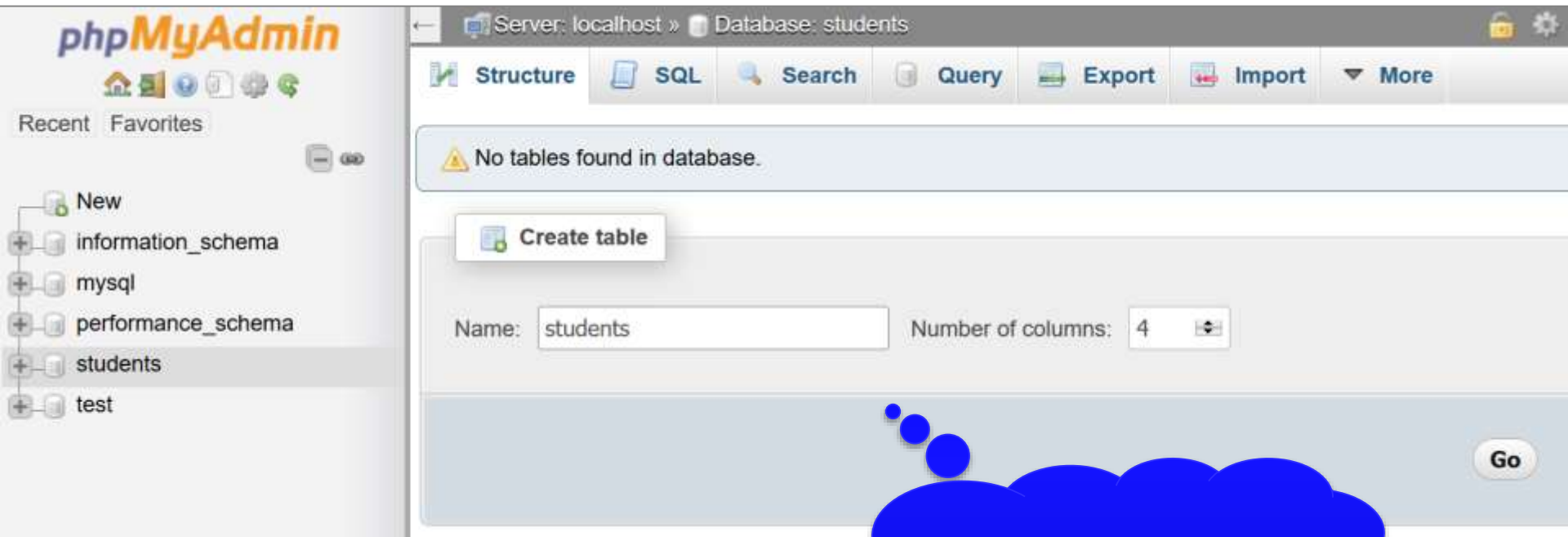
```
(infoiasi)$ mysql -u root mysql
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE,
DROP ON students.* TO 'tux'@'localhost' IDENTIFIED BY
'p@r0la' WITH GRANT OPTION;
Query OK, 0 rows affected (0.11 sec)
```


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 – www.phpmyadmin.net



creăm tabela

Server: localhost » Database: students » Table: students

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table name: students Add 1 column(s) Go

Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
name	VARCHAR	50	None			<input type="checkbox"/>	---
year	ENUM	'1','2','3'	None			<input type="checkbox"/>	---
Edit ENUM/SET values							
id	INT	11	None			<input type="checkbox"/>	PRIMARY
age			None		UNSIGNED	<input type="checkbox"/>	---

Table comments:

PARTITION definition:

Partition by:

Partitions:

Console

Collation:

Storage Engine:

A 4-byte integer, signed range is -2,147,483,648 to 2,147,483,647, unsigned range is 0 to 4,294,967,295

column list)

DATE
Numeric
TINYINT
SMALLINT
MEDIUMINT
INT
BIGINT
-
DECIMAL
FLOAT
DOUBLE
REAL
-
BIT
BOOLEAN
SERIAL
Date and time
DATE
DATETIME
TIMESTAMP

specificăm
structura

Browser Structure SQL Search Insert Export Import Privileges More

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	name	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	year	enum('1', '2', '3')	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	4	age Primary	int(11)	UNSIGNED	No	None			Change Drop More

☐ Check all With selected: Browse Change Drop Primary

Print Propose table structure Move columns Improve table structure

Add 1 column(s) after age Go

+ Indexes

Console Options History Clear

Press Ctrl+Enter to execute query

```
> CREATE TABLE `students`.`students` ( `name` VARCHAR(50) NOT NULL , `year` ENUM('1','2','3') NOT NULL , `id` INT(11) ...  
> CREATE TABLE `students`.`students` ( `name` VARCHAR(50) NOT NULL , `year` ENUM('1','2','3') NOT NULL , `id` INT(11) ...  
> ALTER TABLE `students` CHANGE `id` `id` INT(11) NOT NULL AUTO_INCREMENT  
>
```

vizualizarea
structurii
tabelei create

Column	Type	Function	Null	Value
name	varchar(50)			<input type="text" value="Tuxy Pinguinesscool"/>
year	enum	--		<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3
id	int(11)			<input type="text" value="33"/>
age	smallint(2) unsigned			<input type="text" value="21"/>
<div><input type="checkbox"/> Ignore</div>				
Column	Type	Function	Null	Value
name	varchar(50)			<input type="text"/>
year	enum			<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3
id	int(11)			<input type="text"/>
age	smallint(2) unsigned			<input type="text"/>

-
- ABS
- ACOS
- ASCII
- ASIN
- ATAN
- BIT_LENGTH
- BIT_COUNT
- CEILING
- CHAR_LENGTH
- CONNECTION_ID
- COS
- COT
- CRC32
- DAYOFMONTH
- DAYOFWEEK
- DAYOFYEAR
- DEGREES
- EXP
- FLOOR
-

eventual, inserăm
o înregistrare

Navigation bar with icons and labels: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, More.

✓ Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

`SELECT * FROM `students``

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	name	year	id	age
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Tuxy Pinguinesscool	2	33	21

vizualizarea
datelor existente

php: bd – extensia mysqli

```
// instanțiem obiectul mysqli
$mysql = new mysqli ('localhost', 'tux', 'p@r0la', '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

```
// instanțiem obiectul mysqli
$mysql = new mysqli ('localhost', 'tux', 'p@r0la', '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');
}
```

Parola e dată „în clar”!

► **atenție la probleme de securitate ce pot surveni**

php: bd – extensia mysqli

```
// generăm o listă numerotată cu datele despre studenți
// (cod HTML în stil spaghetti – practică nerecomandată!)
echo ('<ol>');
// rezultatele sunt disponibile într-un tablou asociativ
while ($inreg = $rez->fetch_assoc ()) {
    // coloană a tablei ≡ cheie a tabloului
    echo ('<li>Studentul ' . $inreg['name'] .
        ' este în anul ' . $inreg['year'] . '</li>');
}
echo ('</ol>');

// închidem conexiunea cu serverul MySQL/MariaDB
$mysql->close ();
```

php: bd

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

DBAL – *DataBase Abstraction Layer*

abordare uzuală:

PDO (*PHP Data Objects*)

aspecte pragmatice în tutorialul phpdelusions.net/pdo

// datele de conectare la serverul de baze de date MySQL/MariaDB

\$host = '127.0.0.1';

\$db = 'students';

\$user = 'tux';

\$pass = 'p@r0la'; // atenție: parola e specificată „în clar”!

\$charset = 'utf8';

// stabilirea numelui sursei de date: DSN (*Data Source Name*)

\$dsn = "mysql:host=\$host;dbname=\$db;charset=\$charset";

// opțiuni vizând maniera de conectare

\$opt = [

// erorile sunt raportate ca excepții

PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,

// rezultatele vor fi disponibile într-un tablou asociativ

PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,

// conexiunea e persistentă

PDO::ATTR_PERSISTENT => TRUE

];

php.net/manual/en/book.pdo.php

```
// preluăm de la clientul Web anul de studii (implicit: 2)
$year = $_REQUEST['year'] ? $_REQUEST['year'] : 2;

try {
    $pdo = new PDO ($dsn, $user, $pass, $opt); // instanțiem un obiect PDO

    // pregătim comanda SQL parametrizată
    $sql = $pdo->prepare ('SELECT year, name, age FROM students
                          WHERE year=? ORDER BY age');

    if ($sql->execute ([$year])) { // comanda SQL poate fi executată?
        while ($row = $sql->fetch()) { // ...preluăm fiecare înregistrare găsită
            // ...și o afișăm (coloana tabelului e cheie a tabloului asociativ)
            echo '<p>' . $row['name'] . ' e în anul ' . $row['year'] . '</p>';
        }
    }
} catch (PDOException $e) {
    echo "Eroare: " . $e->getMessage(); // mesajul excepției survenite
};
```

php: bd – extensia pdo

Un posibil rezultat oferit de execuția
programului PHP invocat via un URL precum

<http://profs.info.uaic.ro/~busaco/php/pdo-test.php?year=2>

Tuxy Pinguinesscool e în anul 2

Grace Hopper e în anul 2

Margaret Hamilton e în anul 2

de studiat *script*-ul din
arhiva aferentă prelegerii

php: bd

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

exemple:

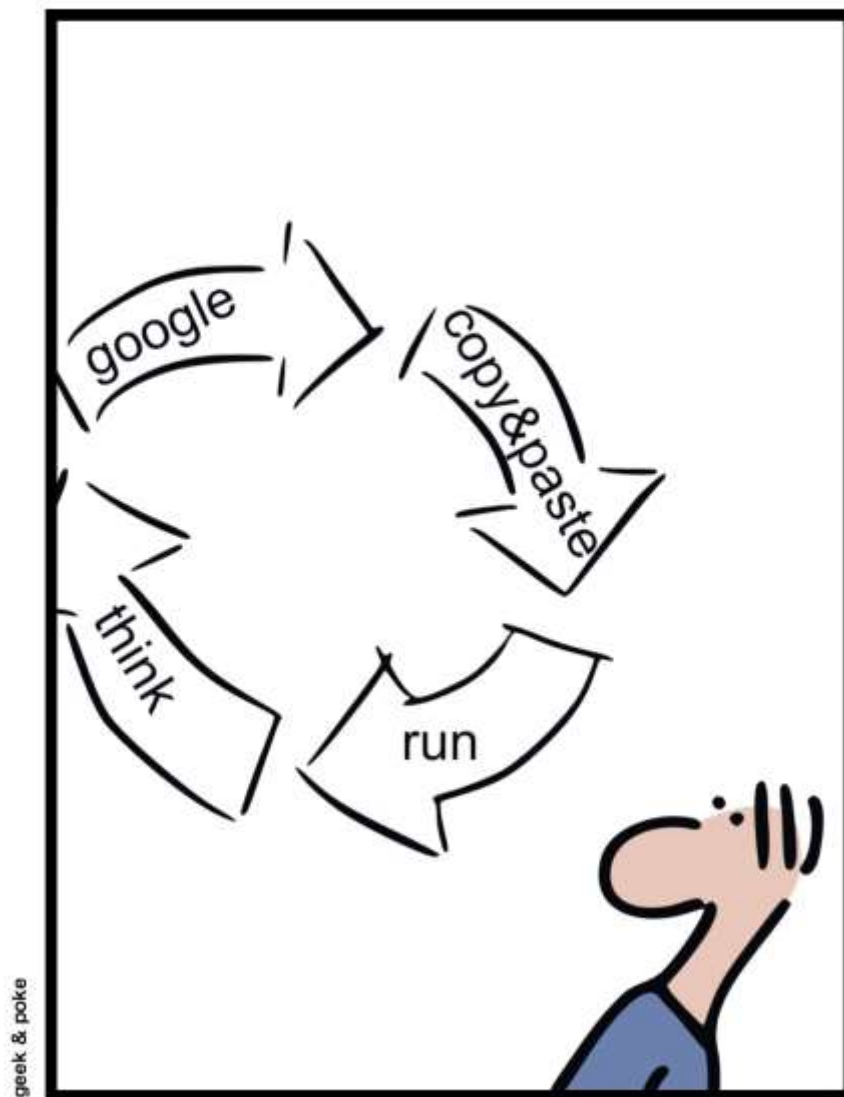
Doctrine – www.doctrine-project.org

Propel – propelorm.org

RedBeanPHP – redbeanphp.com

(în loc de) pauză

SIMPLY EXPLAINED



DEVELOPMENT CYCLE

Instrumente utile pentru dezvoltatorii Web?

pear Register

Search for in the Packages

Main Support Documentation **Packages** Package Proposals Developers Bugs

List Packages Search Packages Statistics Channels

Packages (265) Filter by [PHP 4](#) **PHP 5+** [All PHP Versions](#)

Authentication 4 Auth_RADIUS, Auth_SASL2, OpenID	Caching 1 Cache_Lite	Configuration 1 Config_Lite
Console 5 Console_Color2, Console_CommandLine, Console_GetoptPlus, Console_Table, ...	Database 17 DB_DataObject, DB_Sqlite_Tools, MDB2, MDB2_Driver_fbsql, ...	Date and Time 10 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 17 Archive_Tar, File_Bittorrent2, File_Cabinet, File_CSV_DataSource, ...
File System 1	Gtk2 Components 7	HTML 11

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

HTTP 7 HTTP2, HTTP_FloodControl, HTTP_Header2, HTTP_OAuth, ...	Images 9 Image_3D, Image_Barcode2, Image_Color2, Image_Graph, ...	Internationalization 2 I18N_UnicodeNormalizer, I18N_UnicodeString
---	--	--

Documentation:

- [Support](#)

Downloads:

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

Authentication (5)

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

Caching (13)

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

Console (5)

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

Date and Time (4)

[date_time](#), [hrtime](#), [timecop](#), [timezonedb](#) »

Event (5)

Benchmarking (4)

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

Configuration (4)

[augeas](#), [env](#), [yaconf](#), [zookeeper](#) »

Database (58)

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

Encryption (6)

[gnupg](#), [libsodium](#), [mccrypt](#), [mccrypt_filter](#) »

File Formats (16)

PECL (*PHP Extension Community Library*)

extensii oferite de terți: pecl.php.net

HTML (2)

[html_parse](#), [tidy](#)

Images (13)

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

Languages (8)

[handlebars](#), [lua](#), [mustache](#), [perl](#) »

Mail (3)

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

Multimedia (5)

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

Numbers (1)

[Bitset](#)

HTTP (5)

[apfd](#), [json_post](#), [pecl_http](#), [txforward](#) »

Internationalization (8)

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

Logging (1)

[SeasLog](#)

Math (6)

[big_int](#), [geospatial](#), [hdr_histogram](#), [lapack](#) »

Networking (35)

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

Payment (6)

[coin_acceptor](#), [cybercash](#), [cybermut](#), [mcve](#) »

instrumente: *framework-uri*

Facilități:

MV* și diverse șabloane de proiectare

acces la baze de date (ORM, DAO, ActiveRecord,...)

validare și filtrare a datelor de intrare

autentificare + controlul accesului

management de *cookie*-uri și sesiuni Web

instrumente: *framework-uri*

Facilități:

machete de prezentare a datelor – *templating*

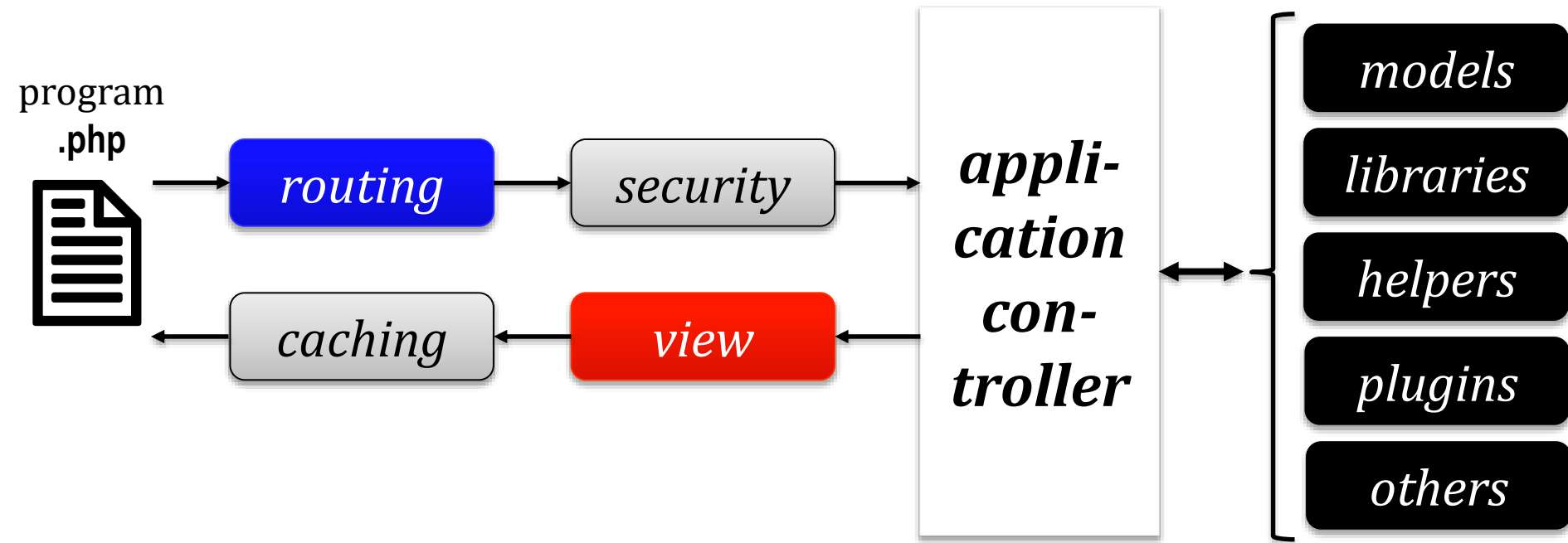
suport pentru performanță – i.e. *caching*

transfer asincron de date (Ajax, WebSocket)

suport pentru servicii Web și API-uri REST

extensibilitate – *e.g.*, module create de programator

instrumente: *framework-uri*



fluxuri de activități realizate de un *framework* Web

instrumente: *framework-uri*

CakePHP – cakephp.org

CodeIgniter – www.codeigniter.com

FuelPHP – fuelphp.com

Kohana – kohanaframework.org

Laravel – laravel.com

Nette – nette.org

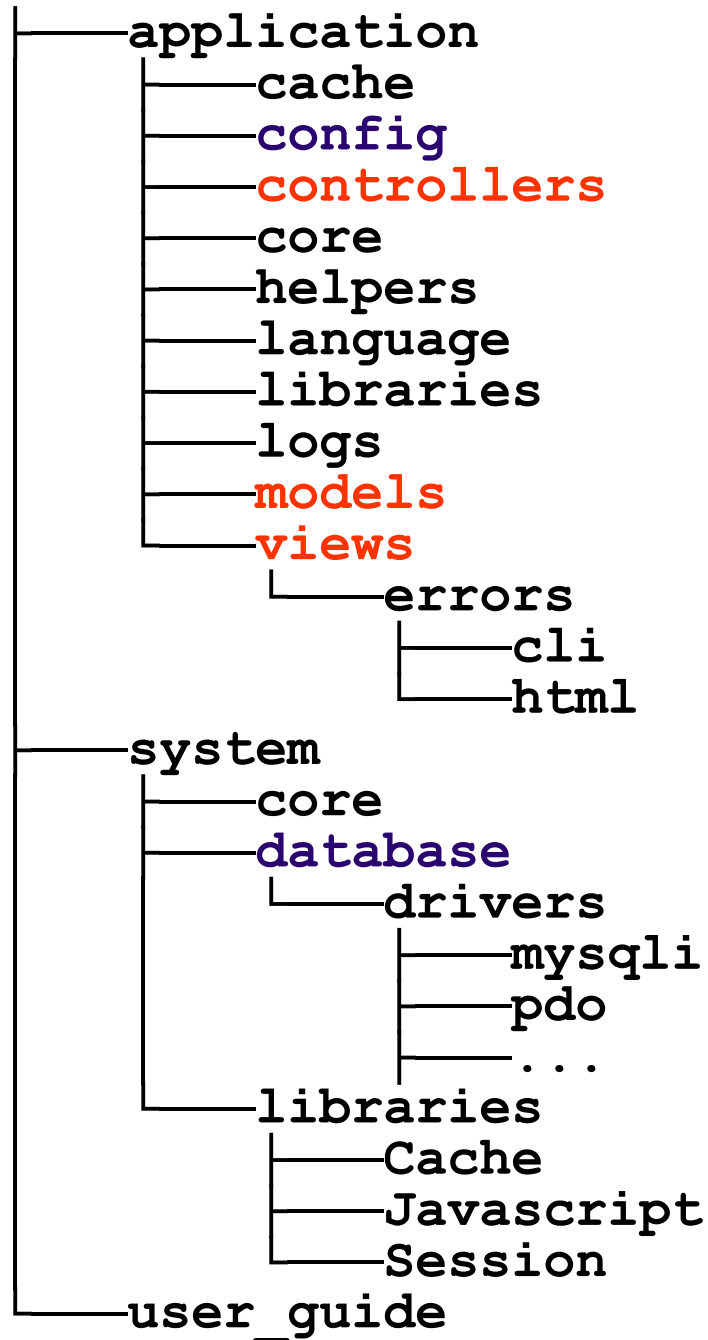
Phalcon – phalconphp.com

PRADO – www.pradoframework.net

Symfony – symfony.com

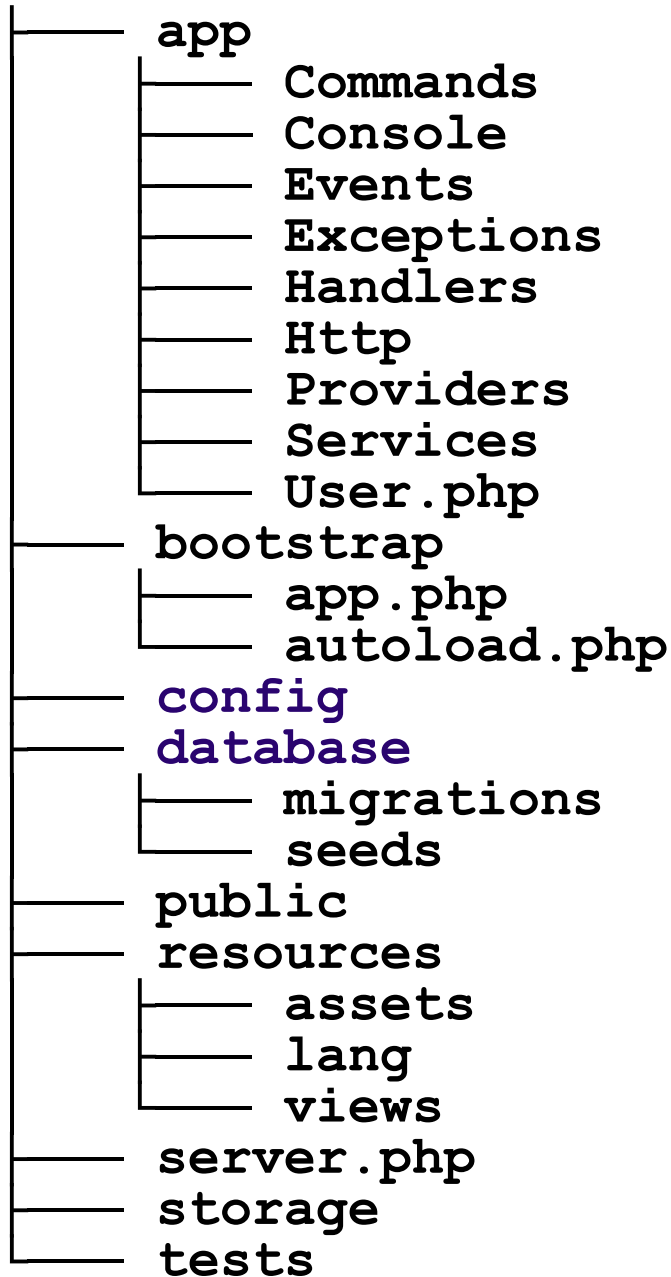
Yii – www.yiiframework.com

Zend Framework – framework.zend.com



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

CodeIgniter
codeigniter.com/docs



structura de directoare
a unei aplicații Web dezvoltate
via *framework*-ul **Laravel**
laravel.com/docs/

instrumente: *micro-framework-uri*

Un *micro-framework* reprezintă
un cadru de lucru Web minimalist

instrumente: *micro-framework-uri*

Un *micro-framework* reprezintă
un cadru de lucru Web minimalist

nu include facilități sofisticate

deseori, focalizat asupra unui singur aspect vizând
dezvoltarea Web – *e.g.*, crearea unui API, microserviciu,...

instrumente: *micro-framework-uri*

Expressive – zendframework.github.io/zend-expressive/

Fat-Free – fatfreeframework.com

Flight – flightphp.com

Lumen – lumen.laravel.com

Slim – www.slimframework.com

Silex – silex.sensiolabs.org

instrumente: pachete

Managementul dependențelor
dintre biblioteci și pachete

Composer
getcomposer.org

www.phptherightway.com/#dependency_management



cookie

avansat

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

phpfastcache/phpfastcache

PHP ↓ 465 998

★ 1 904

PHP Abstract Cache Class - Reduce your database call using cache system. PhpFastCache handles a lot of drivers such as Apc(u), Cassandra, CouchBase, Couchdb, Mongoddb, Files, (P)redis, Leveldb, Memcach

Package type

beagle-component	1
cakephp-plugin	5
component	6
concrete5-package	2
contao-bundle	3
contao-module	5
craft-plugin	11
drupal-custom	1
drupal-library	1

spatie/laravel-cookie-consent

PHP ↓ 192 524

★ 520

Make your Laravel app comply with the crazy EU cookie law

dflydev/fig-cookies

Cookies for PSR-7 HTTP Message Interface

nette/http

Nette Http: abstraction for HTTP request, response and session. Provides careful data sanitization and utility for URL and cookies manipulation.

★ 126

zendframework/zend-session

PHP ↓ 4 145 474

★ 33

manage and preserve session data, a logical complement of cookie data, across multiple page requests by the same client

Tags

Search for other...

<input type="checkbox"/>	cookie	161
<input type="checkbox"/>	cookies	50
<input type="checkbox"/>	session	44
<input type="checkbox"/>	http	30

Packagist
depozit de pachete (*repository*)
gestionate prin **Composer** **packagist.org**

instrumente: medii de dezvoltare

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

Apache + PHP7 +
MySQL/MongoDB +
Perl/Python + ...

AMPPS

www.ampps.com

XAMPP

www.apachefriends.org

oferă inclusiv suport
pentru diverse
configurări de
sisteme Web (extensii)

<https://www.apachefriends.org/add-ons.html>

Applications

Install your favorite apps on top of XAMPP. Bitnami provides a free all-in-one tool to install Drupal, Joomla!, WordPress and many other popular open source apps on top of XAMPP.



WordPress
Blog



Joomla!
CMS



CMS Made Simple
CMS



Drupal
CMS



MediaWiki
Wiki



PrestaShop
e-Commerce



Moodle
eLearning



ownCloud
Media sharing



SugarCRM
CRM



Magento
e-Commerce



Zurmo
CRM



TestLink
Continuous
Integration



Download WinNMP Installer - downloads 9k/month

Latest Version: 18.10 (updated 5 days ago)

WinNMP - Nginx MariaDB Redis Php 7 development stack for Windows

A lightweight, fast and stable server stack for developing php mysql applications on windows, based on the excellent webserver Nginx. A lighter alternative to XAMPP and WAMP, with Composer, Adminer, LetsEncrypt certificates, WinSCP, multiple PHP versions, projects and virtual servers.

Current Package contains:

- Nginx 1.15.5 web server
- MariaDB 10.3.9 database server, mysql 5.5.5 replacement (32/64bit)
- Redis 4.0 Cache/NoSql, memcached alternative (64bit)
- Php 5.6.38 & PHP 7.0.32 & PHP 7.2.10 scripting language (32/64bit)
- XDebug, GeoIP, Gender PHP Extensions
- WinSCP SFTP client
- HTTPS using free LetsEncrypt certificates
- Composer dependency manager for php
- Adminer web based database manager

Nginx + PHP7 + MariaDB +
Redis + unelte de administrare
WinNMP – winnmp.wtriple.com

The screenshot displays a cloud development environment 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' as the active file. The code editor contains a PHP script that prints six messages: 'Mesajul #1' through 'Mesajul #6'. To the right of the code editor, a preview window shows the rendered output of the script, displaying the six messages in a list. At the bottom, a terminal window shows the command 'php/index.php' being executed, with the output 'Starting Apache httpd, serving https://demo-project-busaco.c9.io/php/index.php. Started apache2'. The interface also includes a 'Run' button and a 'Revert' button.

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

Mesajul #1
Mesajul #2
Mesajul #3
Mesajul #4
Mesajul #5
Mesajul #6

Command: php/index.php
Runner: Apache httpd (PHP, HTML)
CWD
ENV

Starting Apache httpd, serving <https://demo-project-busaco.c9.io/php/index.php>.
Started apache2

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

instrumente: execuție online

Editarea și execuția *online* a programelor PHP

Ideone – ideone.com

PhpFiddle – phpfiddle.org

New

New Folder

Save As

Update

Run - F9

FormatAll

Edit

Analyze

Export

Edit

Undo Ctrl-Z

Redo Ctrl-Y

Select all Ctrl-A

Autoformat Selected Alt-Q

Increase indent Ctrl-]

Decrease indent Ctrl-[

PHP autocomplete Alt-1

JavaScript autocomplete Alt-2

MySQL autocomplete Alt-3

jQuery v3.0 autocomplete Alt-4

Comment HTML code

Uncomment HTML code

```

1  <?php
2  /**
3   * PDO MySQL initial code
4   *
5   * User permissions of database
6   * Create, Alter and Index table, Create view, and Select
7   *
8   * @package      PhpFiddle
9   * @link          http://phpfiddle.org
10  * @since        2012
11  */
12  require_once "dBug!.php";
13  require "util/public_db_info.php";
14  $connect = new PDO($dsn, $user_name, $pass_word);
15  //get all tables in the database
16  //$sql = "SHOW TABLES";
17  //get column information from a table in the database
18  //$sql="SELECT COLUMN_KEY, COLUMN_NAME, COLUMN_TYPE FROM
19  //SQL statement for a table data
20  $sql = "SELECT * FROM books WHERE id <= 10";
21  $result = $connect->prepare($sql);
22  //bind parameter(s) to variable(s)
23  //$result->bindParam( . . . );
24  $status = $result->execute();
25  //Next line is same with code from line 29 to 34 for one-time query
26  //$result = $connect->query($sql);
27  if (($status) && ($result->rowCount() > 0))
28  {
29      $results = array();
30      //convert query result into an associative array
31      while ($row = $result->fetch(PDO::FETCH_ASSOC))
32      {
33          $results[] = $row;
34      }

```

le data

WHERE TABLE_NAME

← → ↺ 🏠 sqlfiddle.com/#!9/a6c585/1 ... ♥ ☆

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL

MySQL 5.6
Oracle 11g R2
PostgreSQL 9.6
PostgreSQL 9.3
SQLite (WebSQL)
SQLite (SQL.js)
MS SQL Server 2017

```
1 -- borrowed from https://stackoverflow.com/q/7745609/808921
2
3 CREATE TABLE docs (
4   'id' int(6)
5   'rev' int(3)
6   'content' varchar(255)
7   PRIMARY KEY (id, rev)
8 ) DEFAULT CHARACTER SET utf8
9 INSERT INTO docs ('id', 'rev', 'content') VALUES
10 ('1', '1', 'The earth is flat')
11 ('2', '1', 'One hundred angels can dance on the head of a pin')
12 ('1', '2', 'The earth is flat and round')
13 ('1', '3', 'The earth is like a ball')
14
15
```

```
1 -- based on answer https://stackoverflow.com/a/7745609/808921
2
3 SELECT a.id, a.rev, a.content
4 FROM `docs` a
5 INNER JOIN (
6   SELECT id, MAX(rev) rev
7   FROM `docs`
8   GROUP BY id
9 ) b ON a.id = b.id AND a.rev = b.rev;
```

pentru teste vizând bazele de date relaționale,
de experimentat **DB Fiddle** – www.db-fiddle.com
și/sau **SQL Fiddle** – sqlfiddle.com

Build Schema Edit Fullscreen Browser [;] Run SQL Edit Fullscreen [;]

id	rev	content
1	3	The earth is like a ball.
2	1	One hundred angels can dance on the head of a pin

✓ Record Count: 2; Execution Time: 11ms

id	rev	content
1	3	The earth is like
2	1	One hundred angels can dance on the head of a pin

de studiat și *Industry-specific Data Models*
www.databaseanswers.org/data_models/
SQL Zoo – sqlzoo.net

instrumente: documentare

Generarea automată a documentației unei aplicații

ApiGen – www.apigen.org

Daux – daux.io

phpDocumentor – www.phpdoc.org

phpDox – phpdox.de

Sami – github.com/FriendsOfPHP/Sami

github.com/ziadoz/awesome-php#documentation

instrumente: documentare

Marcaje (adnotări) speciale în comentariile PHP:

@author	@var	@example
@category	@global	@source
@version	@method	@uses
@copyright	@package	@used-by
@license	@subpackage	@link
@see	@param	@internal
@todo	@return	@property
@since	@throws	@property-read
@deprecated	@inheritdoc	@property-write

instrumente: analiza codului

Analiza statică a codului-sursă PHP

pentru descoperirea de greșeli de programare (*bugs*),
verificarea adoptării unor stiluri de redactare
(*coding standards*), corectare automată (*fixers*),
determinarea de metrici (*metrics*):
complexitate, linii de cod,...

github.com/exakat/php-static-analysis-tools

instrumente: analiza codului

Detected violations with PSR2 standards

FOUND 6 ERROR(S) AFFECTING 4 LINE(S)

```
-----
 5 | ERROR | Whitespace found at end of line
 8 | ERROR | Inline control structures are not allowed
 8 | ERROR | Whitespace found at end of line
10 | ERROR | Whitespace found at end of line
13 | ERROR | Expected 1 blank line at end of file; 0 found
13 | ERROR | A closing tag is not permitted at the end of a PHP file
-----
```

PSR – *PHP Standards Recommendations*

*Basic Coding Standard, Coding Style Guide, Logger Interface,
Autoloading Standard, Caching Interface,
HTTP Message Interface, Hypermedia Links etc.*

www.php-fig.org/psr/

instrumente: **testare**

atoum – atoum.org

Codeception – codeception.com

ParaTest – github.com/brianium/paratest

Peridot – peridot-php.github.io

PHPUnit – phpunit.de

SimpleTest – github.com/simpletest/simpletest

altele la github.com/ziadoz/awesome-php#testing

instrumente: testare

Profiling – analizarea și raportarea fragmentelor de cod care se execută lent

PhpBench – github.com/phpbench/phpbench

Tracy – tracy.nette.org

Xdebug extension for PHP – xdebug.org/docs/profiler

XHGui (bazat pe XHPProf) – github.com/perftools/xhgui

Z-Ray – www.zend.com/en/products/server/z-ray

github.com/ziadoz/awesome-php#debugging-and-profiling

instrumente: integrare continuă

CircleCI – circleci.com

PHPCI – www.phptesting.org

Sismo – sismo.sensiolabs.org

github.com/ziadoz/awesome-php#continuous-integration

instrumente: **acceleratoare**

Caching pentru fişiere sursă sau *opcode*-uri + altele

Zend Opcache – integrat în PHP 5.5+
php.net/manual/en/book.opcache.php

de asemenea, de experimentat:

APCu (*Alternative PHP Cache + User Cache*)
Windows Cache for PHP

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

hacklang.org

critici majore

mulți ani: lipsa unei specificații formale a limbajului
actualmente: github.com/php/php-langspect

inconsistență – *e.g.*, foreach, nume de funcții predefinite

lipsa suportului nativ pentru Unicode (exceptând PHP 7)

lipsa suportului nativ pentru *multi-threading*,
însă posibil via extensii ca pthreads

eev.ee/blog/2012/04/09/php-a-fractal-of-bad-design/
www.quora.com/Criticism-of-PHP

Order of arguments (array/string search)

[« Previous Sadness](#)[Sadness Index](#)[Next Sadness »](#)

Decide on an argument order!

```
int strpos ( string $haystack , mixed $needle [, int $offset= 0 ] )  
  
string strstr ( string $haystack , mixed $needle [, bool $before_needle = false ] )  
  
bool in_array ( mixed $needle , array $haystack [, bool $strict ] )  
  
mixed array_search ( mixed $needle , array $haystack [, bool $strict ] )
```

Significance: Consistency

Language consistency is very important for developer efficiency. Every inconsistent language feature means that developers have one more thing to remember, one more reason to rely on the documentation, or one more situation that breaks their focus. A consistent language lets developers create habits and expectations that work throughout the language, learn the language much more quickly, more easily locate errors, and have fewer things to keep track of at once.

critici majore

În PHP se pot ușor crea aplicații
ce „adoptă” *anti-pattern*-ul *Spaghetti Code*


```

<h1>My Users <a class="btn btn-primary" href="new_user.php" role="button">New User</a></h1>
<table class="table table-condensed">
  <tr>
    <th>Id</th>
    <th>Name</th>
    <th>Age</th>
    <th>Email</th>
    <th>Action</th>
  </tr>
  <?php

```

interfața
Web
HTML

PHP pentru acces
la date via SQL
și procesări

```

// Get all users
$stmt = $dbh->prepare("SELECT * FROM users");
$stmt->setFetchMode(PDO::FETCH_ASSOC);
if ($stmt->execute()) {
    while ($row = $stmt->fetch()) {
        ?>

```

```

    <tr>

```

```

        <td><?php echo $row['id']; ?></td>
        <td><?php echo $row['name']; ?></td>
        <td><?php echo $row['age']; ?></td>
        <td><?php echo $row['email']; ?></td>

```

```

    <td>

```

```

        <div class="btn-group" role="group" aria-label="...">

```

```

            <a href="edit.php?id=<?php echo $row['id']; ?>" class="btn btn-default btn-sm">Edit</a>

```

```

            <a href="index.php?delete=<?php echo $row['id']; ?>" class="btn btn-default btn-sm">Delete</a>

```

```

        </div>

```

```

    </td>

```

```

</tr>

```

```

<?php

```

PHP pentru
prezentarea datelor

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
en.wikipedia.org/wiki/Wikimedia_Foundation

studiu de caz: wikipedia

MediaWiki – sistemul *wiki* utilizat pentru toate serviciile implementat în PHP (~70%) + JavaScript (~30%)

MariaDB (soluția principală de stocare)

ImageMagick, DjVu, TeX, rsvg, ploticus etc.

(pentru procesare de conținuturi grafice în MediaWiki)

Apache HTTP Server + NGINX (servere Web)

se oferă inclusiv un API destinat dezvoltatorilor Web:

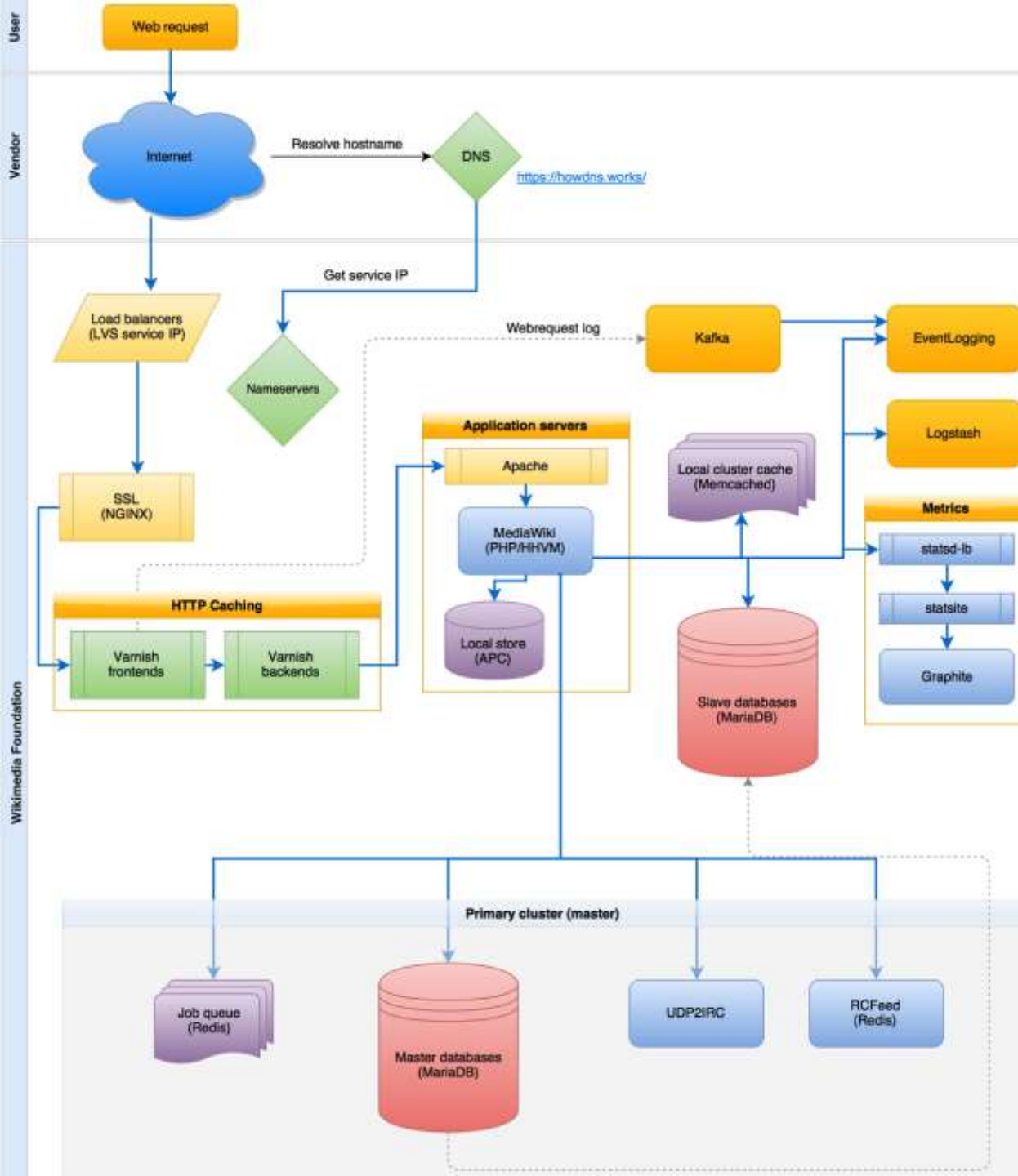
www.mediawiki.org/wiki/API:Main_page

studiu de caz: wikipedia

asigurarea performanței:

- Varnish** (*proxy + caching* pentru conținut HTML)
- Memcached** (*caching* interogări asupra bazelor de date)
- Elasticsearch** (căutare textuală – implementare Java)
- Swift** (stocare redundantă distribuită de la OpenStack)
- gdnssd** (soluție C++ pentru DNS)
- Linux Virtual Server – LVS** (*load balancing*)
- Debian / Ubuntu Server** (sisteme de operare)

meta.wikimedia.org/wiki/Wikimedia_servers



infrastructura
Wikipedia (2015)

studii de caz

Numeroase situri Web recurg la sisteme de management al conținutului (*CMS – Content Management System*) concepute în PHP

generale:

Drupal, Joomla, WordPress etc.

What runs sciencenews.org?  











CMS	Widgets
 Drupal 7	 AddThis
Analytics	
 Hotjar	 Facebook
Font Script	Comments and Reviews
 Font Awesome	 Disqus
 Google Font API	Web Server
	 Apache 2.4
Cache	Programming Language
 Varnish	 PHP 5.6.33
Sales and Marketing	Advertising
 Sumo	 Facebook Pixel

CMS-urile pot include extensii + teme vizuale













CMS	Widgets
 WordPress 4.8.4	 Facebook
Theme Digital-pro-sassy	Video
Plugins Advanced Responsive Video Embedder, Arve Pro, Shiftnav Pro, Views, Types	 MediaElement.js
Analytics	Web Server
 Google Analytics UA	 Nginx 1.2.2
Font Script	Programming Language
 Font Awesome	 PHP 5.6.13

inspectarea tehnologiilor folosite de o aplicație Web cu instrumentul **WhatRuns**

What runs kdnuggets.com?

CMS	Widgets
 WordPress 4.7.9	 AddThis
Analytics	Comments and Reviews
 Google Analytics UA	 Disqus
Web Server	Cache
 Apache 2.4.16	 WordPress Super Cache
Programming Language	Operating System
 PHP 5.3.29	 UNIX
Web Server Extensions	Javascript Frameworks
 OpenSSL 1.0.1e	 jQuery 1.6.2

inspectarea tehnologiilor
folosite de o aplicație Web
cu instrumentul **WhatRuns**

CMS	Widgets
 Joomla	 AddThis
Analytics	Comments and Reviews
 Google Analytics UA	 Disqus
Font Script	Web Framework
 Font Awesome	 Bootstrap
 Google Font API	Web Server
Miscellaneous	 Apache 2.2.15
 K2	CDN
Programming Language	 CloudFlare
 PHP	 CDN JS

studii de caz

Numeroase situri Web recurg la sisteme de management al conținutului (*CMS – Content Management System*) concepute în PHP

de tip *wiki*:












DokuWiki, MediaWiki, pmWiki etc.

studii de caz

Numeroase situri Web recurg la sisteme de management al conținutului (*CMS – Content Management System*) concepute în PHP

specifice comerțului electronic:
Magento, OpenCart, PrestaShop,...

studii de caz

CMS	Widgets	
 PrestaShop	 Facebook	
Hosting Panel	 Google Plus	
 Plesk	 Twitter	
Issue Tracker	<th>Analytics</th>	Analytics
 Bugherd	 Google Analytics UA	
Web Server	<th>Font Script</th>	Font Script
 Apache 2.4.29	 Google Font API	
Advertising	<th>Programming Language</th>	Programming Language
 Twitter Ads	 PHP 5.6.33	













CMS	Widgets	
 Magento 2	 Facebook	
Analytics	<th>Font Script</th>	Font Script
 Google Analytics UA	 Typekit	
Web Framework	<th>Web Server</th>	Web Server
 ZURB Foundation	 Nginx 1.2.1	
Programming Language	<th>Advertising</th>	Advertising
 PHP 5.3.29	 Facebook Pixel	
Tag Managers	<th>Communication</th>	Communication
 Google Tag Manager	 Zendesk Chat	

inspectarea tehnologiilor folosite de o aplicație Web
cu instrumentul **WhatRuns**

studii de caz

Numeroase situri Web recurg la sisteme de management al conținutului (*CMS – Content Management System*) concepute în PHP

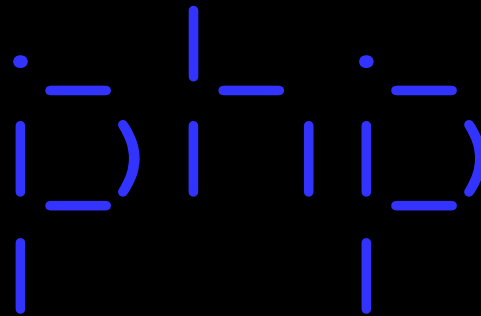
facilitând discuții online (*message board*, forum Web):
bbPress, esoTalk, phpBB,...

What runs forum.softpedia.com?  	
Message Board	Gallery
 IPB	 Lightbox
Analytics	Miscellaneous
 Google Analytics UA	 Google Code Prettify
 Gemius	Programming Language
Web Server	 PHP 7.0.26
 Apache 2.4.29	Databases
Sales and Marketing	 MySQL
 InternetCorp	Tag Managers
Advertising	 Google Tag Manager

inspectarea tehnologiilor folosite de o aplicație Web
cu instrumentul **WhatRuns**

rezumat

privire generală asupra PHP



caracterizare, facilități, instrumente, exemple

```
<event uri="https://stagiipebune.ro/">  
  <name xml:lang="ro">Stagii pe Bune</name>  
  <year>2019</year>  
</event>
```

```
<participant>  
  <name uri="mailto:tux@info.uaic.ro">  
    Tuxy Pinguinnesscool</name>  
  <year kind="Bachelor">2</year>  
</participant>
```

episodul viitor:
un model de date pentru Web: familia XML