

PROGRAMARE WEB

CURS 6- 06.04.2021

Titular: Șef. Lucr. Dr. Mat. Cărbureanu Mădălina

Copyright@Departamentul de Automatică, Calculatoare și Electronică

Universitatea Petrol-Gaze din Ploiești

LIMBAJUL PHP

- Caracteristici și elemente de bază;
- Tipuri de date, variabile și constante;
- Exemple scripturi:
 - Script PHP pentru afișarea unui tabel care conține valorile *sin*, *cos*, *tg* și *ctg* pentru trei unghiuri;
 - Script PHP pentru preluarea datelor dintr-un formular și calculul sumei dintre două numere întregi;
- Aplicații propuse.

Caracteristici și elemente de bază

- **PHP** ➡ limbaj de programare destinat Internetului, *open-source* și *server-side*;
- **PHP** ➡ succesorul aplicației create de programatorul danez Rasmus Lerdof (1994);
- **PHP** ➡ fondatorii companiei Zend Technologies din Israel, Zend Surasky+ Andi Gutmans (1998);
- **PHP** ➡ *Personal Home Page* ➡ *Hypertext Preprocesor*.

Caracteristici și elemente de bază

- **Limbajul PHP:**

- este gratuit;
- este ușor de învățat și de folosit deoarece a împrumutat foarte multe elemente din limbajele C și Pearl;
- poate fi utilizat pe o mare varietate de platforme, precum: Unix, Linux, Windows, Mac OS X;
- dispune de interfețe pentru mare parte din sistemele de gestiune a bazelor de date utilizate în Internet, precum: Oracle, Microsoft SQL Server, MySQL;
- poate fi utilizat împreună cu diferite servere de Web, precum: Apache, IIS-Internet Information Server, iPlanet;
- introduce dinamismul mult dorit în aplicațiile Web prin folosirea codului PHP.

Caracteristici și elemente de bază

- Programe PHP ➡ *scripturi*;
- Script ➡ translatarea codului se realizează prin interpretare;
- Alte limbaje pentru dezvoltare scripturi: Java Script, JSP (JavaServerPages) și ASP (Active Server Pages);
- Scripturile PHP editate în fișiere care au extensia *.php*;
- Orice script PHP este cuprins între etichetele HTML `<?php` și `?>`;
- **Obs:** numai textul cuprins între cele două etichete este interpretat de serverul PHP (XAMPP, WAMP, etc.) și apoi transformat în cod HTML; codul obținut+codul HTML din exteriorul tag-urilor `<?php` și `?>` este transmis browserului, care în urma interpretării acestuia afișează pagina Web corespunzătoare.

- **Obs:**

- instrucțiunea PHP care produce *output* într-o pagină HTML este *echo*:

echo "Text:", \$variabilă_PHP;

- în acest sens se poate utiliza și instrucțiunea *print*, instrucțiune care însă depinde de browserul Web;
exemplu de utilizare instrucțiune *print*:

print "suma dintre a=\$a și b=\$b este s=\$s";

- limbajul PHP nu este *case-sensitive* (deci nu face diferența între literele mari și cele mici) în ceea ce privește cuvintele rezervate ale limbajului (instrucțiunile - de exemplu dacă se scrie *for* respectiv *FOR* limbajul recunoaște instrucțiunea repetitivă cu contor), respectiv funcțiile (fie că este vorba despre cele predefinite ale limbajului, fie că este vorba despre cele definite de către utilizator);
- limbajul PHP este *case-sensitive* în ceea ce privește numele de variabile (variabila *\$a*, respectiv variabila *\$A*, reprezintă două variabile diferite);
- o variabilă PHP se declară conform sintaxei:

\$nume_variabilă=valoare;

Tipuri de date, variabile și constante

- Tipurile de date în limbajul PHP sunt următoarele:
 - tipuri de date scalare: *integer*, *float*, *boolean* (false/False, true/True), *string*;
 - tipuri de date compuse (*array*, *object*);
 - tipuri de date speciale (*resource*, *NULL*).
- În limbajul PHP, numele unei *variabile* este precedat de caracterul "\$", conform sintaxei:

\$nume_variabilă=valoare_memorată;

- Tipul valorii memorate de către variabilă (*valoare memorată*) stabilește tipul variabilei PHP.

Tipuri de date, variabile și constante

- variabilele în limbajul PHP nu se declară la început, ci tipul lor este definit (implicit, de către limbaj) atunci când acestea sunt folosite;
- variabilele în limbajul PHP își schimbă tipul în funcție de valoarea pe care acestea o memorează;
- numele unei variabile în limbajul PHP este dependent de capitalizare (case-sensitive) (de exemplu, *\$a* și *\$A* sunt două variabile diferite);
- variabilele în PHP trebuie să fie inițializate în mod explicit cu o anumită valoare;
- un script PHP poate conține oricâte variabile este nevoie;
- o variabilă PHP poate conține la rândul ei mai multe variabile diferite ce vor fi concatenate folosind simbolul (.) și simbolul (') conform următoarei sintaxe:

\$nume_variabilă=\$var1.' ' \$var2;

- pentru a afișa valoarea unei variabile se va utiliza instrucțiunea *echo*, conform următoarei sintaxe:

***echo "Valoarea variabilei este:", \$nume_variabilă, "
;"***

Tipuri de date, variabile și constante

Tabelul 1. Variabile globale predefinite

Denumire variabilă	Descriere
<code>\$_SESSION</code>	Conține variabile care corespund sesiunii curente a scriptului PHP.
<code>\$_GET</code>	Conține variabilele primite de script prin intermediul unor transferuri care folosesc metoda <code>get()</code> .
<code>\$_POST</code>	Conține variabilele primite de script prin intermediul unor transferuri care folosesc metoda <code>post()</code> .
<code>\$_GLOBALS</code>	Pot fi accesate toate variabilele globale accesibile scriptului.
<code>\$_SERVER</code>	Conține variabilele ale căror valori sunt setate de serverul Web; valorile acestor variabile depind de mediul de execuție al scriptului PHP.
<code>\$_REQUEST</code>	Conține variabilele disponibile prin intermediul oricărui tip de mecanism prin care utilizatorul poate introduce date.

Tipuri de date, variabile și constante

- Declaraire constantă în PHP: *define ("nume_constantă", valoare);*
- **Obs:**
 - diferența între o variabilă PHP și o constantă în PHP este aceea că numele constantei nu este prefixat de caracterul "\$";
 - pentru a afișa valoarea unei constante se va utiliza instrucțiunea *echo*, conform următoarei sintaxe:

*echo "Valoarea constantei este:", nume_constantă, "
";*

Tipuri de date, variabile și constante

Tabelul 2. Constante predefinite

Denumire constantă	Descriere
<code>_LINE_</code>	Returnează numărul liniei dintr-un fișier.
<code>PHP_VERSION</code>	Indică versiunea limbajului PHP folosit.
<code>Get_defined_constants()</code>	Furnizează numele și valorile tuturor constantelor (definite sau predefinite) la care programatorul are acces.

Tipuri de date, variabile și constante

- Tipurile de date compuse *array* și *object*

- Tipul *array* este utilizat pentru reprezentarea *vectorilor*, vectori care în limbajul PHP sunt de fapt mulțimi formate din așa numitele *chei*, chei (de tip *integer* sau *string*) cărora li se asociază anumite valori.
- **Obs:** dacă valorile unui vector sunt scalare, atunci vectorul se numește *unidimensional*, iar în cazul în care sunt de tip *array*, vectorul se numește *multidimensional*.
- Un prim mod de reprezentare (fără utilizarea *cheilor*) al unui *vector unidimensional* este:
$$\text{\textit{\$nume_vector}}=\text{\textit{array}}(\text{\textit{val_1}}, \text{\textit{val_2}}, \text{\textit{val_3}}, \dots, \text{\textit{val_n}});$$
- Un al doilea mod de reprezentare (cu utilizarea *cheilor*) al unui *vector unidimensional* este:
$$\text{\textit{\$nume_vector}}=\text{\textit{array}}(\text{\textit{cheie_1}}\Rightarrow\text{\textit{val_1}}, \text{\textit{cheie_2}}\Rightarrow\text{\textit{val_2}}, \dots, \text{\textit{cheie_n}}\Rightarrow\text{\textit{val_n}});$$

Tipuri de date, variabile și constante

- Crearea unui *vector unidimensional* de *string-uri* se realizează conform sintaxei:

\$nume_vector=array ("șir_1", "șir_2", "șir_3", ... , "șir_n");

- *Vectori unidimensionali asociativi* în care cheile de tip *string* sunt folosite pentru a exprima legătura care există între numele de cheie și valoarea memorată în vector;
- Crearea unui astfel de *vector unidimensional asociativ* se realizează conform sintaxei:

\$nume_vector=array ("nume_cheie_1"=>"val_1", "nume_cheie_2"=>"val_2",..., "nume_cheie_n"=>"val_n");

Tipuri de date, variabile și constante

- **Obs:**
- dacă valorile cheilor unui vector sunt *booleene* (*True* sau *False*), acestea sunt convertite la valorile întregi 1 sau 0;
- *cheie_1, cheie_2, ..., cheie_n* reprezintă cheile de identificare a valorilor *vectorului unidimensional*;
- *val_1, val_2, ..., val_n* reprezintă valorile care se memorează în *vectorul unidimensional* însoțite eventual de chei (de tip *integer*, *string* sau *mixte*);
- o cheie de tip *integer* indică poziția valorii în cadrul unui vector;
- prezența cheilor nu este obligatorie;
- dacă toate cheile lipsesc, celor *n* valori li se asociază de la stânga la dreapta indicii 0, 1, 2, ... , n-1;
- cheile unui vector pot fi mixte (și de tip *integer* și de tip *string*).

Tipuri de date, variabile și constante

- Referirea la valorile unui *vector unidimensional* fără chei și respectiv afișarea valorilor vectorului se realizează folosind instrucțiunea *echo*, conform următoarei sintaxe:

```
echo "Primul elem. al vectorului este:", $nume_vector[0], "<br>;
```

.....

```
echo "Ultimul elem. al vectorului este:", $nume_vector[n-1], "<br>;
```

- Referirea la valorile unui *vector unidimensional* ce utilizează chei și respectiv afișarea valorilor vectorului se realizează folosind instrucțiunea *echo*, conform următoarei sintaxe:

```
echo $nume_vector[cheie_1];
```

.....

```
echo $nume_vector[cheie_n];
```

Tipuri de date, variabile și constante

- Atunci când valorile unui vector sunt de tip *array*, așa cum am menționat, vectorul este unul *multidimensional*;
- Crearea unui *vector multidimensional* se realizează conform următoarei sintaxe:

```
$nume_vector=array(componentă_1=>array("nume_cheie_1"=>"val_1",...,"nume_cheie_n"=>"val_n"),
```

.....

```
componentă_n=>array("nume_cheie_1"=>"val_1",...,"nume_cheie_n"=>"val_n"));
```

Tipuri de date, variabile și constante

- **Obs:** pentru a înțelege modul de construcție a unui *vector multidimensional*, se va analiza vectorul prezentat în tabelul 3.

Tabelul 2. Crearea unui vector multidimensional

Componentă	nume_cheie_1	nume_cheie_2	nume_cheie_n
1	val_1	val_2	val_n
n	val_1	val_2	val_n

Observații:

- referirea la elementele unui *vector multidimensional* se realizează utilizând instrucțiunea *echo*, conform următoarei sintaxe:

*echo \$nume_vector[componentă_i]["nume_cheie_i"], "
", unde*
i=1,..., n.

- **tipul compus de date *object*** permite utilizarea facilităților (obiecte, clase, constructori, moștenire, redefinire metode, etc.) oferite de programarea orientată pe obiecte (POO), facilități ce vor fi discutate ➡ Curs *Clase și obiecte în PHP*.

Exemple scripturi

- Script PHP pentru afișarea unui tabel care conține valorile sin, cos, tg și ctg pentru trei unghiuri;

Tabelul 4. Structură tabel sociat script

Unghi	sin	cos	tg	ctg
10^0	0.1736	0.9848
20^0	0.3420	0.9396
30^0	0.5	0.8660

Script PHP pentru afișarea unui tabel care conține valorile sin, cos, tg și ctg pentru trei unghiuri

<pre><?php \$a=10; \$b=20; \$c=30; \$a1=sin(deg2rad(\$a)); \$a2=cos(deg2rad(\$a)); \$a3=tan(deg2rad(\$a)); \$a4=\$a2/\$a1; \$b1=sin(deg2rad(\$b)); \$b2=cos(deg2rad(\$b)); \$b3=tan(deg2rad(\$b)); \$b4=\$b2/\$b1; \$c1=sin(deg2rad(\$c)); \$c2=cos(deg2rad(\$c)); \$c3=tan(deg2rad(\$c)); \$c4=\$c2/\$c1;</pre>	<pre>echo "<table border='10' cellspacing='1' cellpadding='2' bgcolor='green'> <th>unghi</th> <th>sin</th> <th>cos</th> <th>tg</th> <th>ctg</th> <tr><td>\$a</td> <td>\$a1</td> <td>\$a2</td> <td>\$a3</td> <td>\$a4</td> </tr></pre>	<pre><tr><td>\$b</td> <td>\$b1</td> <td>\$b2</td> <td>\$b3</td> <td>\$b4</td> </tr> <tr><td>\$c</td> <td>\$c1</td> <td>\$c2</td> <td>\$c3</td> <td>\$c4</td> </tr> </table>"; ?></pre>
---	---	---

Script PHP pentru preluarea datelor dintr-un formular și calculul sumei dintre două numere întregi

formular.php

```
<html><body>

<form action="formulartest.php" method="post">
  Introduceți valoarea pentru variabila a:
  <input type="text" name="a" maxlength="5">
  <br>
  <br> <br>
  Introduceți valoarea pentru variabila b:
  <input type="text" name="b" maxlength="5">
  <br>
  <br> <br>
  <input type="submit" value="Calculează suma dintre a și b:">
</form></body></html>
```

formulartest.php

```
<?php
$a=$_POST['a']; $b=$_POST['b'];
echo "Valoarea var a preluată din formular.php este:
",$a,"<br><br><br>";
echo "Valoarea var b preluată din formular.php este:", $b,
"<br><br><br>";
$c=$a+$b; echo "Suma dintre a și b este:", $c, "<br><br><br>";
?>
```

Aplicații propuse

- Testare aplicații 2.2.1 → 2.2.4, pag. 85-89;
- Aplicații propuse: 2.3.1, 2.3.2 (pag. 98) și 2.3.3 (pag. 100);
- **Obs:** se va utiliza cartea “*Programare web. Ghid teoretic și practice*”, autor Șef lucr. dr. mat. Cărbureanu Mădălina, Editura Universității-Petrol Gaze din Ploiești, 2020.

Spor la lucru!