

## Ce este PHP ?

PHP este abrevierea de la PHP – Hypertext Preprocessor .

PHP este un limbaj de programare pentru creerea site –urilor dinamice si interactive si a fost introdus de Rasmus Lerdorf in 1994. De atunci PHP a suferit numeroase schimbari.

In termeni tehnici PHP este un limbaj tip script, cross platform, HTML embedded si server side.

**Cross platform** – puteti rula PHP pe sisteme de operare diferite, in majoritatea cazurilor fara modificari.

**HTML embedded** – codul PHP este o mixtura de instructiuni PHP si cod HTML.

**Server side** – programele scrise in PHP ruleaza pe un server (in general un Web Server cum ar fi Apache, IIS, etc..).

Programele PHP sunt o mixtura de trei elemente : *text, cod HTML si script PHP*

## I INTRODUCERE

### 1. Scrierea scheletului programelor PHP

Fiecare program PHP include două linii speciale, care indică serverului PHP că textul cuprins între cele două linii este alcătuit din instrucțiuni PHP. Practic, aceste linii pot fi asimilate copertelor unei cărți, care pastrează unitatea programului dumneavoastră PHP. Între aceste două linii vor fi scrise instrucțiunile PHP. Pentru a începe să scrieți un program PHP, deschideți editorul dumneavoastră de texte și introduceți următoarele două linii :

```
<?php
```

```
....  
?>
```

- Acesta este un cod de început și nu afișează nimic.

Apoi, salvați scriptul dumneavoastră sub formă de fișier text, în directorul "www" din "wamp", cu un nume care respectă următoarele reguli:

Este recomandat ca numele fișierului să fie alcătuit numai din caractere minuscule, cifre și liniute. Utilizarea de spații, majuscule și alte caractere ar putea crea probleme pe sisteme de operare diferite.

Extensia numelui fișierelor trebuie să fie .php.

### 2. Afișarea datelor de ieșire într-un browser Web

Programele PHP execută trei categorii de operații elementare:

- a) Obține date de la un utilizator.

- b) Executa prelucrari ale datelor, respectiv obtine accesul la datele stocate in fisiere si baze de date si le manipuleaza.
- c) Afiseaza date astfel incat un utilizator sa le poata vizualiza.

Primele doua operatii sunt oarecum mai dificil de realizat decat cea de-a treia. Totusi, afisarea datelor astfel incat acestea sa fie vizibile utilizatorului este o operatie foarte simpla. Asa cum paragrafele unui text scris sunt compuse din propozitii, programele PHP sunt alcatuite din instructiuni. Regulile care controleaza formarea propozitiilor se numesc sintaxa. Acelasi termen este folosit si pentru a desemna regulile care guverneaza formarea instructiunilor PHP.

Iata un exemplu pentru crearea instructiunii PHP care trimite date de iesire la un browser Web, astfel incat acestea sa fie vizibile pentru un utilizator:

```
<?php  
echo "scrieti aici un text oarecare";  
?>
```

Observati ca instructiunea incepe cu un cavant "**echo**" si se incheie cu un caracter punct si virgula (;).

Constructia **echo** trimite datele de iesire care vor fi afisate de browser.

Ghilimelele duble se folosesc pentru delimitarea unei expresii de tip text, in cazul nostru "scrieti aici un text oarecare" (se pot folosi si ghilimele simple).

In locul propozitiei "scrieti aici un text oarecare" puteti plasa aproape orice text, cu exceptia altor ghilimele. Totusi, pentru moment, trebuie sa includeti numai litere, cifre, spatii si semne de punctuatie folosite in alfabetul latin, precum virgula, caracterul punct si virgula, punctul, semnul de intrebare si semnul exclamarii. De asemenea, puteti include caracterele < >, folosite pentru delimitarea etichetelor HTML, respectiv caracterul /, folosit pentru a indica membrul de inchidere al unei perechi de etichete HTML.

De exemplu, iata o instructiune PHP care are drept date de iesire un fragment dintr-un vers din Scrisoarea a III-a de Eminescu:

```
<?php  
echo "<h2> Iata vine-un sol de pace...</h2>";  
?>
```

- Ca sa vedeti rezultatul, salvati fisierul cu acest cod in directorul www din "wamp", de exemplu cu denumirea teste.php si apelati in browser

Perechea de etichete H2 determina formatarea datelor de iesire ca titlu HTML de nivel 2.

Scripturile PHP pot fi incluse si in fisiere cu cod HTML, ca in urmatorul exemplu (dar salvate cu extensia ".php"):

```
<html>  
<head>  
<title>Test PHP</title>  
</head>  
<body>  
<?php  
echo "<h4> Exemplu de script PHP inclus in HTML</h4>" ;  
?>
```

```
</body>
</html>
```

In browser va apare :

Exemplu de script PHP inclus in HTML

Pot fi incluse si scripturi mai complicate ce contin zeci sau chiar sute de linii, atata timp cat acestea sunt incluse intre cele doua linii specifice limbajului PHP:

```
<?php
?>
```

### **3. Documentarea unui script PHP**

In afara de a furniza nume descriptive fisierelor ce contin scripturile dumneavoastra PHP, e necesar sa includeti in interiorul fiecarui script atat comentarii care sa permita unui cititor sa determine cu usurinta utilitatea scriptului, cat si alte informatii referitoare la script. De exemplu, puteti include un comentariu care precizeaza numele autorului scriptului si ce face acesta.

Iata un model sintactic pentru comentariile PHP:

// Scrieti aici comentariul dumneavoastra (pe o singura linie)

Dupa cum se poate vedea, un comentariu incepe cu doua caractere slash, urmate de un spatiu. In continuare, linia contine comentariul dumneavoastra, care poate include orice caractere doriti, inclusiv caractere speciale.

Iata un exemplu simplu de script PHP care include comentarii:

```
<?php
// Acest script afiseaza un mesaj vizibil pentru utilizator.
echo "Acesta este un script foarte simplu.";
?>
```

**Nota:** Fiecare linie a comentariului trebuie sa inceapa cu doua caractere slash //. Totusi, puteti crea un comentariu din mai multe linii si in alte moduri. Iata un exemplu:

```
/*
```

*Acesta este un comentariu pe mai multe linii.*

*Poate fi alcatuit dintr-un numar oricat de mare de linii.*

```
*/
```

Pentru a adauga un comentariu alcatuit din mai multe linii, scrieti la inceput caracterele /\*, iar la sfarsit trebuie sa se incheie cu caracterele \*/. Intre cele doua perechi de caractere, puteti scrie orice text doriti, folosind oricate linii doriti.

### **5. Executarea unui script PHP**

Dupa ce ati creat un script PHP, veti dori sa-l executati.

Sa luam un alt exemplu, ce contine cod HTML si comentarii:

Deschideti editorul dumneavoastra de texte si introduceti urmatoarele linii :

```
<html>
<head>
<title>Test PHP</title>
</head>
```

```

<body>

<?php
// Script test
// Acest script afiseaza un mesaj vizibil pentru utilizator.
echo "Primul meu script PHP.";
?>
</body>
</html>

```

Salvati fisierul cu numele test-script.php  
 Salvati fisierul in directorul www al serverului.  
 Deschideti browserul si ACCESATI test-script.php

## II Variabile, constante, tipuri de date

O *variabilă* este un nume (identificator) care se asociază unei zone de memorie în care pot fi memorate diverse valori. Conținutul zonei de memorie de memorie poate fi schimbat cu alte valori permise, ori de câte ori este nevoie.

Spre deosebire de variabile, constantele sunt entități a căror valoare rămâne neschimbată pe toată durata execuției unui script.

Atât variabilele cât și constantele aparțin unui *tip de date*. Un *tip de date* indică mulțimea valorilor posibile precum și operațiile care se pot face asupra acestor valori.

Numele unei *variabile PHP* trebuie să fie alcătuit din litere, cifre sau liniuțe de subliniere, primul caracter al numelui fiind obligatoriu o literă sau liniuță de subliniere. În plus, *numele trebuie să fie precedat de semnul \$*.

**Exemplu:** \$A, \$a, \$pret, \$\_abd

Numele unei variabile este case sensitive.

În PHP tipul unei variabile nu este declarant explicit, el se schimbă în funcție de tipul valorii memorate.

Definirea unei *constante* se face cu ajutorul funcției predefinite *define()*

**Exemplu:** define('oras', 'ploiesti')

Programele de calculator manipuleaza datele, care reprezinta informatii. Programele PHP folosesc doua categorii principale de date: numere si siruri. Numerele sunt compuse mai ales din cifre, în timp ce un sir poate contine orice caracter, inclusiv cifre, litere si simboluri speciale.

Decizia privind modul de stocare a datelor este importanta, în mod caracteristic, datele se stocheaza sub forma de numere atunci când se doreste executarea unor operatii matematice asupra datelor, deoarece numerele sunt stocate într-un mod care permite efectuarea de calcule. Pe de alta parte, sirurile sunt stocate folosind o modalitate care faciliteaza înțelegerea lor de catre operatorul uman. Datele trebuie stocate sub forma de siruri daca formatul acestora nu este numeric sau daca doriti ca operatorul uman sa fie capabil de a introduce sau de a vizualiza datele. Practic, puteti asimila numerele cu un mod de stocare a datelor în interiorul calculatorului. Sirurile se pot asimila unui mod de stocare a datelor în afara calculatorului.

Aceste doua mari categorii de date: numere si siruri, formeaza opt tipuri principale de variabile :

- **Boolean**
- **Integer**
- **Float**
- **String**
- **Array**
- **Object**
- **Resource**
- **Null**

In mod normal tipul variabilelor nu este specificat explicit; acesta va fi evaluat de catre interpretorul PHP la momentul run-time (in momentul executarii scriptului).

### ***1. Tipul boolean***

Variabilele de tipul boolean pot lua doar doua valori : FALSE sau TRUE

Variabilele de alt tip decat boolean pot fi convertite la tipul boolean prin operatorul **cast (bool)** sau (boolean), plasate inaintea valorilor, desi in general nu e nevoie de aceasta converire, valoarea fiind automat recunoscuta de PHP.

### ***2 . Tipul integer***

PHP foloseste doua categorii de numere: întregi (tipul integer) si duble, cu virgula (tipul float)

Variabilele integer reprezinta numerele întregi, ele pot fi specificate in format zecimal, hexazecimal sau octal.

Numerele întregi reprezinta numerele fara parte fractionara folosite la numarare, plus zero si numerele negative. Cu alte cuvinte, în PHP termenul de întreg are aceeasi semnificatie ca si în matematica. De exemplu, numarul 100 poate fi reprezentat în PHP sub forma de întreg.

Scrierea numerelor PHP este simpla. Un întreg PHP se obtine prin scrierea cifrelor care îi alcatuiesc valoarea. Daca valoarea este negativa, scrieti un semn minus imediat la stânga numarului. Evitati sa scrieti spatii sau virgule ca parte a unui întreg PHP.

Iata câteva exemple de numere PHP întregi :

- 215678 - Numar in reprezentare zecimala
- 0x1A8; - Numar in reprezentare hexazecimala (reprezinta:  $1 \cdot 16 \cdot 16 + 10 \cdot 16 + 8 = 424$  (in zecimal))
- 067 - Reprezinta:  $6 \cdot 8 + 7 = 55$  (in zecimal)

Valorile minime, respectiv maxime pe care le poate lua o variabila de tip integer depind de sistemul de operare pe care ruleaza modulul PHP. De exemplu pentru un sistem Windows valoarea unui integer se memoreaza pe 32 de biti: 31 bit pentru numar si un bit pentru semn. In acest caz valorile unui intreg se afla in intervalul :-231 respectiv: 231. Daca incercam sa folosim un integer cu valori care ies din acest interval vor avea de-a face cu un fenomen de depasire : integer overflow. In acest caz interpretorul PHP converteste acesta valoare intr-o valoare de tip float (care are un interval mai extins de valori).

În PHP simpla împărțire a două valori întregi va produce ca rezultat o variabilă de tip float. Pentru a obține doar partea întreagă acestui rezultat putem folosi operatorul cast (int) în fața rezultatului.

Ex.: `echo (int)8.7;` (va returna 8)

Pentru a rotunji rezultatul la întregul cel mai apropiat de valoarea reală se poate folosi funcția `round()`, unde între paranteze se adaugă valoarea.

Ex.: `echo round(8.7);` (va returna 9)

### 3. Tipul float

Reprezintă variabile de tip real [numerele cu virgulă], (în lb. engleză se folosește punctul zecimal în loc de virgulă). De exemplu 2.5

În general, numerele duble (tipul float) sunt stocate folosindu-se formatul standard IEEE-64, care furnizează 64 de biți. Acest format va permite să stocați valori care pot merge până la  $1,8 \times 10^{308}$  sub forma de numere duble și furnizează aproximativ 14 cifre după punctul zecimal (sau cifre semnificative) de precizie.

Iată câteva exemple de numere duble (tipul float)

123.4567 - Număr de tip float  
1.2e3; - Reprezintă:  $1.2 \times 10^3$

### 4. Tipul string

Reprezintă o înșiruire de caractere, fiecare caracter este memorat pe 1 byte; setul de caractere este limitat la 256 valori distincte. Dimensiunea șirurilor poate fi oricât de mare în PHP, nu există specificații care să limiteze numărul maxim de caractere dintr-un șir.

Spre deosebire de întregi și de numere float, care conțin cifre, șirurile pot conține orice caracter. Ca atare, șirurile sunt utile pentru stocarea datelor care nu pot fi calculate, precum nume, fraze și adrese.

În PHP, un șir poate fi declarat în mai multe feluri:

1) Un mod simplu de a declara un șir este prin delimitarea cu ghilimele simple (' ')

Dacă dorim ca șirul să conțină caracterul ' va trebui să inserăm înaintea lui caracterul escape \. Dacă dorim ca șirul să conțină caracterul \ va trebui să dublăm acest caracter. Acest comportament este exemplificat în exemplul următor:

```
<?php
$var1 = 'Acesta este un șir de test';
echo 'Curs \'PHP\'';
echo '<br />Vrei să stergi C:\*. *?';
echo '<br />Variabila var1=$var1';
echo "<br />Variabila var1=$var1";
?>
```

În browser va apărea:

Curs 'PHP'

Vrei sa stergi C:\\*.\* ?

Variabila var1=\$var1

Variabila var1=Acesta este un sir de test

Observati diferenta dintre ultimile doua linii!

In penultima linie, unde s-a folosit ghilimele simple pentru delimitarea sirului, variabila \$var1 nu este expandata adica nu este afisat valoarea variabilei var1=Acesta este un sir de test ci exact textul scris (numele ei). Pentru a realiza afisarea valorii variabilei si nu numele acesteia, se folosesc ghilimelele duble, precum in ultima linie a exemplului de mai sus.

2) O alta metoda pentru a specifica un sir în PHP este prin folosirea ghiimelelor duble (" ").

Daca intr-un sir delimitat prin ghilimele duble vrem ca acesta sa afiseze numele unei variabile, nu valoarea acesteia, adaugam un caracter backslash (\) inaintea numelui variabilei.

Exemplu; **echo "Numele variabilei este \\$var";**

Acest cod va afisa Numele variabilei este \$var, oricare ar fi valoarea variabilei "\$var".

Caracterele care alcatuiesc sirul sunt incluse între ghilimele duble (" "); De exemplu, sirul reprezentând numele fizicianului care a formulat teoria relativitatii este "Albert Einstein". Asa cum s-a explicat, un sir poate contine date numerice; de exemplu, "3.14159".

Daca se doreste adaugarea de ghilimele duble intr-un sir delimitat de ghilimele duble, se adauga caracterul \ in fata acestora.

Exemplu; **echo "Numele programului este \"PHP\".";**

Prin delimitarea cu ghilimele duble PHP faciliteaza includerea în siruri a unor caractere speciale, precum caracterele de salt la linie noua sau retur de car, prin furnizarea de secvente escape care reprezinta caractere speciale.

Iata secventele escape folosite în PHP:

- \n - salt la linie noua
- \r - retur de car (rand nou)
- \t - caracter de tabulare pe orizontala
- \\ - backslash
- \\$ - simbolul dolarului
- \\" - ghilimele duble

Ca exemplu, iata un sir care include un retur de car, urmat de un salt la linie noua:  
"Salut, lume!\r\n".

Retineti ca fiecare secventa escape începe cu un backslash (\). Pentru a include un backslash într-un sir, trebuie sa folositi doua caractere backslash.

Pe langa imbricarea variabilelor in cadrul sirurilor delimitate prin ghilimele duble, PHP pune la dispozitie operatorul de concatenare a sirurilor: . (punct) . Acest operator adauga un sir la sfarsitul altui sir. De exemplu:

```
<?php
$nume = 'Popescu';
$prenume='Costel';
echo 'Numele de familie este '.$nume. 'iar prenumele este '.$prenume;
?>
```

In browser va apare:

Numele de familie este Popescu iar prenumele este Costel

In anumite situatii este necesar sa accesam unul dintre caracterele unui sir. Pentru aceasta putem folosi parantezele{ } ca in exemplul de mai jos:

```
<?php
$var1 = 'Acesta este un sir de test';
echo $var1{0}; // Afisaza A (primul caracter din sir)
echo $var1{2}; // Afisaza e (al treilea caracter din sir)
?>
```

Rezultatul afisat va fi

Ae

3) Sintaxa heredoc este o alta modalitate de a delimita siruri

In acest caz delimitatorul este ("<<<"); acesta trebuie urmat de un identificator unic, dupa care urmeaza sirul de caractere, iar secventa se incheie din nou cu identificatorul mentionat. Identificatorul de incheiere trebuie sa se afle in prima coloana a liniei, acesta poate contine caractere alfanumerice dar neaparat trebuie sa inceapa cu o litera, nu cu o cifra sau alt semn.

Veti intelege mai bine studiind exemplu de mai jos:

```
<?php
$var1 = <<< EOT
Exemplu de sir care foloseste delimitatorul heredoc.
EOT;
echo $var1;
?>
```

Rezultatul afisat va fi:

Exemplu de sir care foloseste delimitatorul heredoc.

## 5. Tipul array

Reprezinta un vector de valori (care creaza o matrice, vezi Lectia 9 despre matrice), fiecare element al matricei are asociat o cheie. Aceasta cheie va fi folosita ulterior la identificarea unui element specific at matricei. In PHP tipul array se mai numeste si tip map ordonat, deoarece elemente vectorului sunt ordonate dupa campul cheie.

Variabilele de tip array si lucrul cu acestea vor fi prezentate mai detaliat in lectiile urmatoare. Iata un exemplu simplu de variabile array:



```
<?php
$fructe[0] = 'mere';
$fructe[1] = 'caise';
$fructe[2] = 'piersici';
?>
```

## 6. Tipul object

Reprezinta de fapt instanta unei clase declarate in PHP. O clasa este o structura care contine variabile membru si functii membru.

Variabilele de tip object si lucrul cu acestea vor fi prezentate detaliat in lectiile urmatoare.

Pentru a defini un obiect care poate fi folosit pentru afisarea mesajului "Salutare lume!", se scrie urmatoarea

secventa:

```
<?php
class Salutare {
function ArataSalutare() {
echo 'Salutare lume!';
}
}
$a = new Salutare;
echo $a -> ArataSalutare();
?>
```

## 7. Tipul resource

Este un tip special de variabila care pastreaza o legatura spre resurse externe. Exemple de resurse externe: manipulatori pentru deschidere de fisiere, conectare la baze de date, compresia fisierelor, resurse COM, etc...

## 8. Tipul NULL

Reprezinta variabilele care nu au inca atribuita o valoare.

O variabila se considera a avea valoarea Null daca:

- este setata explicit prin atribuirea valorii NULL
- nu a fost asignata inca o valoare acestei variabile
- variabila a fost stearsa prin functia unset();

Putem afla tipul unei variabile folosind functia **gettype()** care returneaza un string (sir) continand tipul variabilei cercetate.

Observati si studiatii exemplul de mai jos

```
<?php
$var1 = TRUE;
$var2 = 100;
$var3 = 23.88;
$var4 = "Nume";
$var[5] = "fructe";
```

```
echo gettype($var1);  
echo '<br />'.gettype($var2);  
echo '<br />'.gettype($var3);  
echo '<br />'.gettype($var4);  
echo '<br />'.gettype($var[5]);  
echo '<br />'.gettype($var6);  
?>
```

In browser va apare:

boolean  
integer  
double  
string  
string  
NULL