

PHP

Hypertext Preprocessor

PHP este un limbaj de programare destinat, in primul rand, Internetului, aducand dinamica unei pagini web. Este unul din cele mai importante limbaje de programare web open-source si server-side. Numele PHP provine din limba engleza si este un acronim recursiv : Php: Hypertext Preprocessor.

Intr-o pagina HTML se poate ingloba cod PHP, care va fi executat la fiecare incarcare a paginii. PHP este interpretat pe serverul WEB si genereaza un cod HTML care va fi vazut de utilizator.

PHP are urmatoarele caracteristici:

- Poate fi utilizat pe toate sistemele de operare majore, inclusiv Linux, multe variante Unix (inclusiv HP-UX, Solaris si OpenBSD), Microsoft Windows, Mac OS X, RISC OS
- Oferă posibilitatea de a alege sistemul de operare si serverul
- Oferă posibilitatea de a utiliza programarea procedurala sau programarea orientata pe obiecte (POO), sau o combinatie a acestora
- Reduce timpul de a crea mari site-uri web
- Deschide mii de posibilitati de instruire on-line.
- Permite crearea de magazine virtuale.

Sintaxa PHP

Sintaxa si semantica sunt similare cu majoritatea limbajelor de programare (C, Java, Perl), cu toate ca adaugarea de cod PHP necesita utilizarea unei etichete. Intotdeauna, codul php este delimitat cu etichetele:

```
<?php
    //cod php
?>
```

SAU

```
<?
    //cod php
?>
```

In PHP comentariile se pot face in felul urmator

```
//comentariu
/* comentariu
    pe mai multe randuri
*/
```

Daca avem inserat cod PHP intr-un fisier HTML si dorim ca broser-ul web sa il interpreteze corect trebuie ca extensia fisierului sa fie php (exemplu: a.php). Daca fisierul nu are extensia *.php, codul PHP va aparea in pagina.

```
<html>
    <head>
        <title>Exemplu</title>
    </head>
    <body>
        <?php
            echo "Salut, acesta este primul meu script PHP";
        ?>
    </body>
</html>
```

Instructiunea **echo** va afisa pe pagina.

Variabile

In PHP, definim o variabila cu urmatoarea forma:

```
$variabila="Aceasta este o variabila";
$var= 4;
```

Variabile compuse din mai multe cuvante trebuie sa fie separate cu `_`: \$variabila_mea. O variabila este construita dintr-un **\$** care se pune in fata numelui, un `=` si ghilimele magice `' '` sau `" "` in interiorul carora se afla valoarea variabilei si apoi constructia este inchisa cu `;`

```
<?php
    $variabila="Aceasta este o variabila";
    echo $variabila;
?>
```

Tipuri de date

String

Înainte de a putea folosi un sir de caractere, trebuie mai întâi să îl creem. Un sir de caractere poate fi folosit direct într-o funcție, sau poate fi pus într-o variabilă. Mai jos vom crea exact același text de două ori: în primul caz vom pune sirul într-o variabilă, iar în cel de-al doilea punem sirul de caractere direct în funcție.

```
<?php
    $string="Buna ziua!";
    echo "Buna ziua!";
    echo $string;

?>
```

În exemplul de mai sus, primul sir de caractere este stocat în variabila `$string`, în timp ce cel de-al doilea sir de caractere va fi folosit în funcția **echo** și nu trebuie să fie depozitat.

Array

Vectorii în PHP sunt niște mulțimi formate din chei. Fiecărei chei din vector i se atasează o valoare. Acest tip de date este optimizat astfel încât să poată fi folosit în locul următoarelor structuri de date: liste, colecții, stive, cozi și altele.

Valoarea unei variabile de tip vector se poate specifica în mai multe moduri, printre care:

```
$array= array("cheie1" => "valoare1", "cheie2"=> "valoare2",...);
```

Sau

```
$array["cheie1"]="valoare1";
$array["cheie2"]="valoare2";    ...
```

Sau

```
$array[]="valoare1";
$array[]="valoare2";    ...
```

Sau

```
$array=array("valoare1","valoare2",...);
```

Există trei tipuri de array:

- Array numeric
- Array asociativ
- Array multidimensional

```
<?php
    $a= array('denumire'=>'portocala',
              'familie'=>'citrice',
              'culoare'=>'portocaliu'
              'gust'=>'dulce'
    );
    echo $a['denumire'].'<br />'.$a['familie'];
?>
```

Array numeric :

```
$nume = array("Ion","Ana","Ghita");
```

Array asociativ:

```
$ani = array("Ion"=>32, "Ana"=>30, "Ghita"=>34);
```

Array multidimensional:

```
$familie=array( "Caini"=>array(  "Ciobanesc German",
                                "Husky Siberian",
                                "Dalmatian"),
               "Pisici"=>array("Siameza",
                              "Russian Blue")
);
```

Un aspect foarte important in PHP este faptul ca un array nu trebuie sa contina un anumit tip de date, el putand fi mixt:

```
$a[0]=1;
$a[1]="aaa";
```

Primul index al unui array este 0, astfel, primul element va fi \$a[0]. Ultimul element al unui array va avea indexul cu o unitate mai mic decat dimensiunea array-ului.

Boolean

Variabilele de acest tip pot avea doar doua valori: ADEVARAT sau FALS. Aceste valori pot fi indicate prin cuvintele cheie **TRUE** sau **FALSE**

Integer

O variabila de tip integer reprezinta o valoare din multimea numerelor intregi. Aceste numere pot fi specificate in baza 10, in baza 16 sau in baza 8, conventiile fiind aceleasi ca si in limbajele C/C++,Java.

Float

O variabila de tip float e specificata folosind fie forma zecimala, fie cea stiintifica (cu exponent).

Operatori

Operatorii de atribuire

Operatorul de atribuire este: "="

```
$ex1= 3;
$ex2= $ex1;
```

Atat \$ex1 cat si \$ex2 vor avea valoarea egala cu 3.

Operatorii aritmetici

Operator	Nume	Exemplu
+	Adaugare	2+2
-	Scadere	4-2
*	Inmultire	3*5
/	Impartire	6/3
%	Modulo	50%5

```
<?php
    $adunare=2+2;
    $scadere=4-2;
    $inmultire=3*5;
    $impartire=6/3;
    $modulo=50%5;

?>
```

Operatori de comparare

Comparatiile sunt utilizate pentru a verifica relatia dintre variabile si/sau valori. Operatorii de comparare sunt folositi in interiorul conditiilor pentru a face evaluarea returnand adevarat sau fals.

Operator	Nume	Exemplu
==	Egal cu	\$x==\$y
!=	Diferit	\$x!=\$y
<	Mai mic decat	\$x<\$y
>	Mai mare decat	\$x>\$y
<=	Mai mic sau egal	\$x<=\$y
>=	Mai mare sau egal	\$x>=\$y

Operatori String

Punctul “.” este utilizat pentru a lipi doua siruri de caractere. In PHP, caracterul punct se numeste operator de concatenare.

```
<?php
    $sir1="Salut ";
    $sir2="George!";
    echo $sir1.$sir2;

?>
```

Instructiunea IF/ELSE

Instructiunea if/else este similara celei din alte limbaje de programare.

```
<?php
    $numar = 3;
    if ( $numar == 3 ) {
        echo " Adevarat! Numarul este 3";
    } else {
        echo "Fals! Numarul nu este 3";
    }

?>
```

Instructiunea **if** verifica conditia. Daca aceasta este indeplinita, se executa codul din prima acolada. In caz contrar, se executa codul din ramura **else**.

Instructiunea ELSEIF

Aceasta instructiune este o combinatie dintre instructiunea **if** si **else**. Aceasta poate verifica fiecare conditie pana in momentul in care una dintre conditiile gasite returneaza o valoare adevarata.

```
<?php
    $a = 20;
    $b = 1;
    $rezultat = $a + $b;
    if($rezultat == '20')
    {
        echo "Rezultatul este egal cu 20";
    }
    elseif ($rezultat == '21') {
        echo "Rezultatul este egal cu 21";
    }
    else {
        echo "Rezultatul nu este egal cu cel din conditie";
    }

?>
```

Instructiunea SWITCH

Aceasta instructiune functioneaza asemanator cu **if**, insa permite conditiilor sa aibe mai mult de doua valori. Intr-o instructiune **if**, conditia poate fi adevarata sau falsa, insa intr-o instructiune **switch** conditia poate lua orice numar de valori diferite. Aceasta instructiune trebuie sa contina o instructiune case care sa manevreze fiecare valoare pe care o dorim.

```
<?php
    $destinatie = "China";
    echo "Calatorind in $destinatie <br />";
    switch ($destinatie){
        case "Bucuresti":
            echo "poti vedea Palatul Parlamentului";
            break;
        case "China":
            echo "poti vedea Zidul Chinezesc";
            break;
    }
?>
```

Valoarea variabilei \$destinatie a fost China, asa incat, atunci cand PHP efectueaza trecerea la switch se face o cautare pentru un case cu valoare "China". Cand este gasit case-ul respectiv este executat codul din acel segment.

Instructiunea WHILE

Cel mai simplu tip de bucla PHP este **while**. Asemenea instructiunii **if**, ea se bazeaza pe o actiune. Diferenta dintre **if** si **while** este aceea ca instructiunea **if**, daca gaseste adevarata conditia, afiseaza o singura data bucata de cod din ea, insa in conditia **while**, daca rezultatul este adevarat, bucata de cod respectiva se va repeta atata timp cat conditia este adevarata.

```
<?php
    $numar = 1;
    while($numar <= 5){
        echo $numar."<br />";
        $numar++;
    }
?>
```

Instructiunea FOR

O alternativa cu o functionalitate mai ridicata pentru utilizarea buclelor este structura repetitiva **for**. Sintaxa este foarte asemanatoare cu cea din limbajele JavaScript, C/C++ si Java si anume:

```
for(expresie1; conditie; expresie2) {  
    //instructiune  
}
```

Prima expresie este evaluata o singura data, inainte de inceperea executiei ciclului. Expresia conditie este testata inaintea fiecarei repetari a buclei. Daca expresia returneaza fals, repetarea se opreste. Expresia 2 este executata la sfarsitul fiecarei repetari. Instructiunea se executa la fiecare repetare a buclei. Oricare dintre cele trei expresii poate lipsi; in cazul in care o expresie lipseste, se considera ca ea are valoarea true.

```
<?php  
    for ($variabila = 1; $variabila <= 10; $variabila++) {  
        echo $variabila.'<br />';  
    }  
?>
```

Structura FOREACH

Structura **foreach** ofera o cale simpla de parcurgere a array-urilor. Exista doua moduri de utilizare a acestei structuri. Prima sintaxa parcurge un array definit de variabila_array. La fiecare iteratie, valoarea elementului curent este atribuita variabilei variabila_element care se poate folosi ca orice variabila simpla.

```
foreach(variabila_array as variabila_element) {  
    //instructiune  
}
```

A doua sintaxa realizeaza acelasi lucru, numai ca indicele elementului din array este atribuit la fiecare iteratie variabilei variabila_cheie.

```
foreach(variabila_array as variabila_cheie=>variabila_element) {  
    //instructiune  
    //variabila_element este, de fapt, variabila_array[variabila_cheie]  
}
```