

PHP- Variabile; Operatori



Violeta Chis

Variabile

In PHP sunt opt tipuri de variabile primitive:

- patru tipuri scalare: întregi, numere cu virgula flotantă, șiruri și booleene
- două tipuri compuse: matrice și obiecte
- două tipuri speciale: resurse și NULL

Variabile

- **Intreg** (integer) este un simplu număr întreg exprimat în sistem zecimal, hexazecimal sau octal, opțional putând purta și semn (+ sau -).
In sistem octal numărul trebuie precedat de 0 iar pentru hexazecimal precedat de 0x.

\$a=2020; //număr exprimat în sistem zecimal

\$a=-12; //număr negativ exprimat în sistem zecimal

Variabile

- **Numărul cu virgula flotanta** (float) sau dublă precizie este un număr real care poate fi folosit cu următoarea sintaxă:

\$a=37.24; //număr zecimal

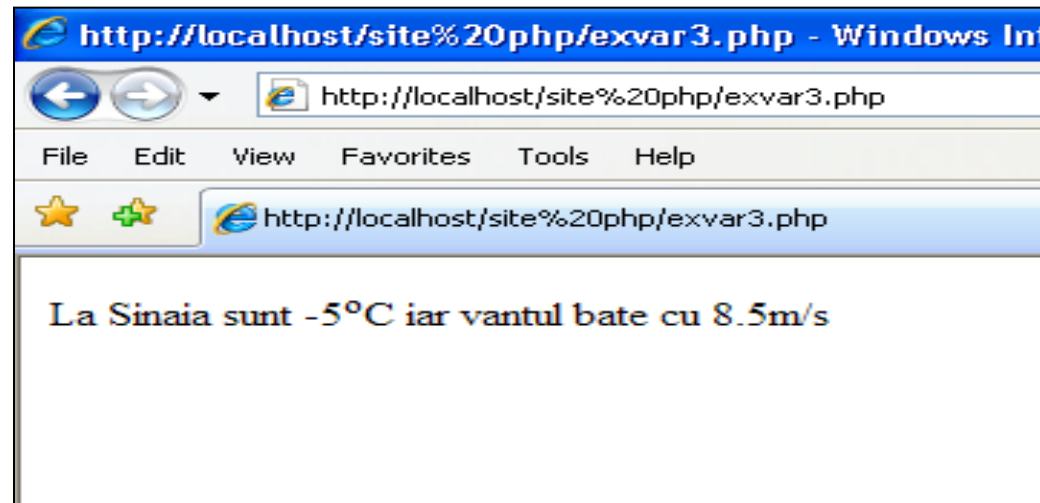
\$a=-5.12; //număr zecimal negativ

\$a=4.2e2; //420

\$a=2E+3; //2000

Variabile

```
<?php  
$temperatura=-5;  
$vant=8.5;  
echo "La Sinaia sunt ".$temperatura."<sup>o</sup>C iar vantul bate  
cu ".$vant."m/s";  
?>
```



Variabile

- **Sirul** (string) este o variabilă care conține o combinație de numere, litere, simboluri și spații delimitate între două ghilimele simple sau duble.
- Folosind ghilimele duble pot fi specificate mai multe caractere speciale, printre care:
 - sfârșit de linie ("r")
 - rând nou ("n")
 - tab orizontal ("t")
 - semnul dolar ("\$")
 - ghilimelele duble ("")
- Șirurile pot conține și nume de variabile.
- interpretorul PHP nu oferă suport nativ pentru setul de caractere Unicode

Variabile

Tipul boolean: TRUE , FALSE

Există posibilitatea de a converti o variabilă de orice tip la tipul *boolean*.

În momentul efectuării unei conversii, sunt convertite la valoarea *FALSE* următoarele valori:

- numărul întreg 0;
- numărul real 0.0;
- șirul vid;
- șirul "0";
- un vector fără nici un element;
- un obiect fără nici o variabilă membru;
- o variabilă de tipul *NULL*;
- o variabilă nedefinită

Orice altă valoare este convertită la valoarea *TRUE* (inclusiv resursele).

Variabile

Tipul array

În PHP, sunt trei tipuri de array-uri:

1. numerice – cu index numeric
2. asociative – fiecare cheie ID key e asociată cu o valoare
3. multidimensională- un array conținând unul sau mai multe arrayuri

Vectorii în PHP sunt niște mulțimi formate din *chei*.

Fiecărei chei din vector i se atașează o valoare.

Valoarea unei variabile de tip vector se poate specifica folosind construcția

`array (cheie => valoare, cheie => valoare, ...)`

Cheile pot avea și valori negative

Nu se pot folosi pentru chei variabile de tipul *array* sau *object*.

Variabile

Array-uri numerice

- stochează fiecare element din array cu un index numeric.
- sunt două metode de creare :

1. index asignat automat (index startează de la 0):

```
$cars=array("Saab","Volvo","BMW","Toyota");
```

2. atribuire de index manual:

```
$cars[0]="Saab"; $cars[1]="Volvo"; $cars[2]="BMW"; $cars[3]="Toyota";
```

```
<?php
$cars[0]="Saab"; $cars[1]="Volvo"; $cars[2]="BMW"; $cars[3]="Toyota";
echo $cars[0] . " and " . $cars[1] . " are Swedish cars.";

?>
```

Rezultat: Saab and Volvo are Swedish cars.

Variabile

Array-uri asociative

- fiecărei chei ID I se asociază o valoare.
- când stocăm date despre valori cu nume specifice, un array numeric nu e întotdeauna cel mai potrivit.
- cu array asociativ putem folosi valori ca și chei și să le atribuim valori.

Exemplul 1

Se atribuie vârste unor persoane:

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

Variabile

Exemplul 2

La fel ca exemplul 1, dar in alt mod:

```
$ages['Peter'] = "32";
```

```
$ages['Quagmire'] = "30";
```

```
$ages['Joe'] = "34";
```

Cheile ID folosite in script astfel :

```
<?php
```

```
    $ages['Peter'] = "32";
```

```
    $ages['Quagmire'] = "30";
```

```
    $ages['Joe'] = "34";
```

```
    echo "Peter are " . $ages['Peter'] . " ani.";
```

```
?>
```

rezultat:

Peter are 32 ani.

Variabile

Array-uri multidimensionale

- fiecare element în array-ul principal poate fi tot un array. Și fiecare element în sub-array poate fi un array, etc.

Exemplu -un array multidimensional, cu chei ID atribuite automat :

```
$famiii = array ("Griffin"=>array (  
    "Peter",  
    "Lois",  
    "Megan" ),  
    "Quagmire"=>array (  
    "Glenn"  
    ),  
    "Brown"=>array (  
    "Cleveland",  
    "Loretta",  
    "Junior" )  
);
```

Variabile

Array-ul arata astfel explicit cu ID-urile elementelor

```
: Array ([Griffin] => Array (
```

```
[0] => Peter
```

```
[1] => Lois
```

```
[2] => Megan
```

```
)
```

```
[Quagmire] => Array (
```

```
0] => Glenn
```

```
)
```

```
[Brown] => Array (
```

```
[0] => Cleveland
```

```
[1] => Loretta
```

```
[2] => Junior ) )
```

Variabile

Tipul object

- Pentru a utiliza o variabilă de tip obiect va trebui să realizăm o *instanțiere* prin intermediul instrucțiunii **new**. Sintaxa este:
\$a = new salutare;
Astfel, variabila **\$a** devine un obiect al cărui metode pot fi utilizate
- Ex definim un obiect care poate fi folosit pentru afișarea mesajului *SALUTARE LUME!* :

```
class salutare
{
function Displaysalutare ()
{
    echo "SALUTARE LUME!";
}
}
```

- Pentru afișarea propriu-zisă a mesajului va trebui să executăm metoda **Displaysalutare()** printr-o instrucțiune de tipul
\$a -> Displaysalutare();

Variabile

Tipul resource

- sunt folosite pentru păstrarea unor referințe către anumite resurse externe cum ar fi conexiuni la baze de date, fișiere, etc.
- sunt create și utilizate de anumite funcții speciale
- valoarea nici unei variabile de alt tip nu poate fi convertită la tipul *resource*

Variabile

Tipul NULL:

- Valoarea specială *NULL* este atribuită oricărei variabile care nu a fost inițializată. Această valoare este singura pe care o pot avea variabilele de tip *NULL*.
- Se consideră că o variabilă are tipul *NULL* dacă:
 - i s-a atribuit constanta *NULL*;
 - nu a fost inițializată;
 - a fost dezinițializată (prin intermediul funcției *unset()*).

Variabile

- Intr-o variabila pot fi mai multe variabile cu continut diferit iar “lipirea” variabilelor se face cu .’ si respectiv ‘. si anume daca avem 2 variabile le vom uni astfel
\$variabila1.’ ‘.\$variabila2

<?php

```
$nume = 'Pop';
```

```
$prenume = 'Radu';
```

```
$nume_complet = $nume.' '.$prenume;
```

```
echo ' Ma numesc '.$nume_complet.' ';
```

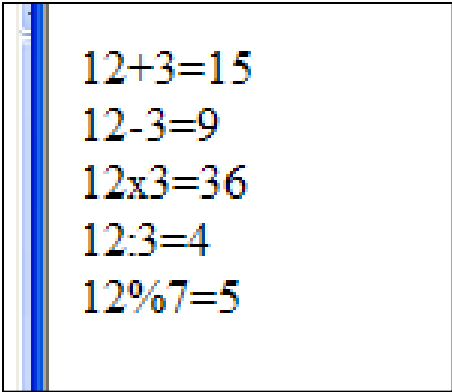
?>

Operatori

- Operatori aritmetici
 - adunare ('+');
 - scădere ('-');
 - înmulțire ('*');
 - împărțire ('/');
 - modulo ('%').

Operatori

```
<?php
$a=12;
$b=3;
$c=7;
$adunare=$a+$b;
$scadere=$a-$b;
$inmultire=$a*$b;
$impartire=$a/$b;
$modulo=$a%$c;
echo $a."+".$b."=".$adunare."<br>";
echo $a."-".$b."=".$scadere."<br>";
echo $a."x".$b."=".$inmultire."<br>";
echo $a."/".$b."=".$impartire."<br>";
echo $a."%".$c."=".$modulo;
?>
```



12+3=15
12-3=9
12x3=36
12:3=4
12%7=5

Operatori

Operatori de comparare

\$a == \$b	egal cu
\$a === \$b	identic cu
\$a != \$b	diferit
\$a <> \$b	diferit
\$a !== \$b	neidentic
\$a < \$b	mai mic
\$a > \$b	mai mare
\$a <= \$b	mai mic sau egal
\$a >= \$b	mai mare sau egal

'?' operatorul condițional - are forma

expresie1?expresie2:expresie3

are ca rezultat valoarea expresiei *expresie2* dacă valoarea expresiei *expresie1* este **true** (eventual după conversia la tipul boolean) sau valoarea expresiei *expresie3* în caz contrar

Operatori de incrementare/decrementare

- ++\$a Pre-incrementare
- \$a++ Post-incrementare
- --\$a Pre-decrementare
- \$a-- Post-decrementare

Operatorul atribuire =

Operatorul de concatenare - .

Operatori

Operatori logici

Operator	Nume	Utilizare	Rezultat
!	NOT	!\$b	Returnează <i>true</i> dacă \$b este <i>false</i> și viceversa
&&	AND	\$a && \$b	Returnează <i>true</i> dacă atât \$a cât și \$b sunt <i>true</i> și <i>false</i> în caz contrar
	OR	\$a \$b	Returnează <i>true</i> dacă \$a, \$b sau ambele sunt <i>true</i> și <i>false</i> în caz contrar
and	AND	\$a and \$b	Identic cu &&, dar are prioritate mai scăzută
or	OR	\$a or \$b	Identic cu , dar are prioritate mai scăzută

Afişarea mesajelor în PHP

- *echo*
- o variantă mai scurtă a lui *echo*, (dacă este setat în *php.ini* directiva *short_open_tag* pe *ON*) :

<?=\$foo?>

- *print* - folosind ghilimelele duble, orice variabilă din cadrul stringului este parsată

Utilizarea fișierelor incluse

- *require(numele_fisierului);*
 - *fișierul de includere* (de obicei are extensia *inc*) - este inserat în script, înlocuind instrucțiunea *require*.
 - dacă fișierul de includere conține linii de program PHP, trebuie să includă etichetele *<?php* și *?>*, amplasate în locațiile adecvate.
 - nu puteți folosi o expresie pentru a specifica numele fișierului care urmează a fi inclus de către o instrucțiune *require*
- *require_once* asigură faptul că fișierul specificat este inclus o singură dată într-un script dat
- *include*

Utilizarea fișierelor incluse

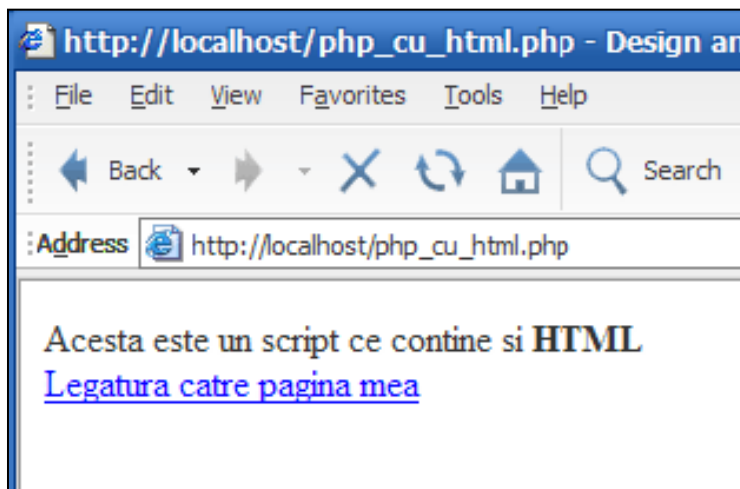
```
<?php
```

```
echo 'Acesta este un script ce contine si
```

```
<b>HTML</b> <br>
```

```
<a href="pagina_mea.php">Legatura catre pagina  
mea</a>';
```

```
?>
```



Bibliografie

- <https://www.tutorialspoint.com/php/index.htm>
- <https://www.php.net/>