

PROGRAMARE WEB

CURS 7- 13.04.2021

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

Copyright@Departamentul de Automatică, Calculatoare și Electronică

Universitatea Petrol-Gaze din Ploiești

LIMBAJUL PHP

- Operatori;
- Funcții pentru șiruri de caractere;
- Exemple;
- Aplicații propuse.

OPERATORI

- **Operatori lb. PHP** → marea lor majoritate similari celor din limbajul C/C++;
- **Operatorii de bază** ai limbajului PHP → Tabelul 1;

OPERATORI

Tabelul 1. Operatori PHP

Operator	Descriere
new	definește un obiect al unei clase
[]	paranteze pătrate
++/--	incrementare/decrementare
~ (NOT pe bit), (int), (float), (string), (array), (object), (bool)	operatori de negație la nivel de bit și conversie între tipuri de date
!	operator de negație logică
*, /, %	operatori aritmetici multiplicativi
+, -, . (operator de concatenare)	operatori aritmetici aditivi și de concatenare stringuri
<<, >>	operatori <i>shift</i> (stânga și dreapta)
<, <=, >, >=, < > (diferit)	operatori de comparație
==, !=, === (operator identitate), !== (operator de neidentitate)	operatori de comparație
&	ȘI pe bit (referință)

OPERATORI

Tabelul 1. Operatori PHP

^	SAU exclusiv bit cu bit
 	SAU bit cu bit
&&	ȘI logic
 	SAU logic
?:	operator condițional (terțiar)
=, +=, -=, /=, *=, .=", &=", =", ^=, <<=", >>="	operatori de atribuire
and	ȘI logic
xor	SAU logic exclusiv
or	SAU logic
,	operator de secvențiere
(tip) expresie	operator <i>cast</i> (permite conversia valorii unei variabile la tipul de date dorit)
::	<i>scope access operator</i> (operator de referire la domeniul de valabilitate a unei funcții membru)

OBSERVAȚII

- în PHP nu există niciun operator pentru efectuarea de împărțiri întregi (de exemplu, rezultatul operației $7/2$ nu este numărul întreg 3, ca în limbajele C/C++, ci rezultatul va fi unul de tip *float*, respectiv 3.5); valoarea reală 3.5 este convertită la valoarea întreagă 3, prin utilizarea de conversii;
- o variabilă de orice tip poate fi convertită la tipul *float*, *boolean* și tipul *string*;
- pentru concatenarea a două șiruri de caractere se va utiliza operatorul "." și nu operatorul "+";
- în PHP, manipularea biților unui întreg sau a codurilor ASCII (*American Standard Code for Information Interchange*) asociate caracterelor unui șir de caractere se realizează prin utilizarea operatorilor la nivel de bit (&, |, ~, <<, >>) ce se regăsesc printre operatorii prezentați în tabelul 1.
- în PHP, în șirurile de caractere pot fi incluse caractere speciale (așa numita *secvență escape*), precum: \n (salt la linie nouă), \r (retur de car-rând nou), \t (caracter de tabulare pe orizontală), \\ (backslash), \\$ (simbolul dolarului) și \" - ghilimele duble.

FUNCȚII PENTRU ȘIRURI DE CARACTERE

- Șiruri de caractere ➡ *strings*;
- Funcții predefinite pentru lucrul cu șiruri de caractere ➡ Tabelul 2;

FUNCȚII PENTRU ȘIRURI DE CARACTERE

Tabelul 2. Funcții pentru *strings*

Nume funcție	Descriere
<i>str_word_count</i> (\$sir)	returnează numărul de cuvinte dintr-un șir de caractere;
<i>str_replace</i> ("șir1", "șir2", "\$sir")	înlocuiește în "\$sir" caracterele specificate la "șir1" cu cele de la "șir2";
<i>chunk_split</i> (\$sir, nr, 'sep')	împarte \$sir în mai multe subșiruri, având lungimea de caractere nr și separate prin separatorul 'sep';
<i>explode</i> ('sep', \$sir)	crează un vector multidimensional (un <i>array</i>) cu părțile din \$sir separate prin separatorul 'sep'; fiecare subșir delimitat de separator, devine un element în vectorul multidimensional;

FUNCȚII PENTRU ȘIRURI DE CARACTERE

Tabelul 2. Funcții pentru *strings*

Denumire variabilă	Descriere
<i>rtrim</i> ("\$\$sir", 'caractere')	șterge din partea dreaptă a șirului caracterele specificate la 'caractere' (un singur caracter sau un șir de caractere); dacă al doilea parametru nu este specificat se va șterge spațiile libere din dreapta;
<i>str_repeat</i> (\$\$sir, <i>nr</i>)	returnează un șir de caractere în care parametrul \$\$sir este repetat de <i>nr</i> ori;
<i>str_shuffle</i> (\$\$sir)	amestecă aleator caracterele din \$\$sir;
<i>strcmp</i> (\$\$sir1, \$\$sir2)	compară cele două șiruri date ca parametri, fără a face diferență între majuscule și litere mici; returnează valoarea 0 dacă cele două șiruri sunt egale, o valoare negativă dacă \$\$sir1 este mai mic decât \$\$sir2 sau o valoare pozitivă dacă \$\$sir1 este mai mare decât \$\$sir2;

FUNCȚII PENTRU ȘIRURI DE CARACTERE

Tabelul 2. Funcții pentru *strings*

Denumire variabilă	Descriere
<i>strpos(\$sir, 'subsir')</i>	returnează poziția de unde începe ' <i>subsir</i> ' în <i>\$sir</i> (începând de la valoarea 0); dacă ' <i>subsir</i> ' nu este găsit, funcția returnează valoarea <i>false</i> ;
<i>strrev(\$sir)</i>	inversează elementele unui șir de caractere;
<i>strtolower(\$sir)</i>	transformă toate literele din șir în litere mici;
<i>substr_count(\$sir, 'subsir')</i>	returnează de câte ori ' <i>subsir</i> ' se găsește în <i>\$sir</i> .

Obs:

- funcțiile *str_replace*, *strpos* și *substr_count* sunt *case-sensitive*.

EXAMPLE

```
<?php
echo "operatori", "<br>";
$a=22;
$b=10;
$a=$b+$c;
echo "suma este", $a, "<br>";
$x=$y=30;
echo "val. a este", $a, "<br>";
echo "val. b este", $b, "<br>";
$v=25;
$s=2;
$w=$v%$s;
echo "restul impartirii este", $w, "<br>";
$x="concatenare";
$y="siruri";
$z=$x.$y;
echo "rezultat operator de concatenate:", $z, "<br>";
$s=2;
$s++;
$t=1;
$t--;
echo "incrementare:", $s, "<br>";
echo "decrementare:", $t, "<br>";
$p=9/4;
$k=(integer) $p;
echo "converie la intreg", $k, "<br>";
?>
```



localhost/aplicatii/test10.php

operatori

suma este10

val. a este10

val. b este10

restul impartirii este1

rezultat operator de concatenate:concatenaresiruri

incrementare:3

decrementare:0

converie la intreg2

EXAMPLE

- Pentru stringul dat (*`$sir="Limbajul PHP este un limbaj de programare destinat Internetului"`*) folosind funcțiile pentru manipularea șirurilor de caractere, să se realizeze următoarele operații:

 - determinarea numărului de cuvinte din șirul de caractere dat;
 - înlocuirea în *`$sir`* a șirului de caractere "*de programare destinat Internetului*", cu șirul de caractere "*open_source*";
 - împărțirea șirului inițial *`$sir`* în mai multe subșiruri, având lungimea de nouă caractere separate prin separatorul "|";
 - crearea unui vector multidimensional ce va conține părțile din șirul *`$sir`* separate prin separatorul ",";
 - ștergerea din partea dreaptă a șirului inițial *`$sir`* a caracterelor "*destinat Internetului*";
 - afișarea de două ori a șirului inițial *`$sir`*;

EXAMPLE

- amestecarea aleatorie a caracterelor din șirul de caractere `$sir`;
- compararea lexicografică a șirului de caractere `$sir` cu șirul de caractere `$sir1= "Limbajul PHP este un limbaj de programare destinat Internetului, open-source și server-side"`;
- returnarea poziției de unde începe subșirul "PHP" în șirul de caractere `$sir`;
- inversarea șirului de caractere `$sir`;
- transformarea tuturor literelor din șirul de caractere `$sir` în litere mici;
- afișarea numărului de apariții ale subșirului "limbaj" în șirul de caractere `$sir`.

EXAMPLE

- REZOLVARE:

```
$sir="Limbajul PHP, este un limbaj de programare, destinat Internetului";
echo "Sir pentru testarea functiilor specializate:", $sir, "<br>";
echo "numarul de cuvinte din sir este:";
echo str_word_count($sir);
echo "<br>";
echo str_replace("de programare destinat Internetului", "open_source", "$sir");
echo "<br>";
echo chunk_split($sir, 9, ' | ');
$array = explode(' ', $sir);
echo "<br>";
print_r($array);
echo "<br>";
echo $sir;
echo "<br>";
echo rtrim("$sir", 'destinat Internetului');
echo str_repeat($sir, 2);
echo str_shuffle($sir);
$sir1="TEST";
$sir2="TEST";
echo "<br>";
if (strcmp($sir1, $sir2) == 0) {
    echo "Sirurile sunt egale";
}
echo strpos($sir, 'PHP');
echo "<br>";
echo "situl inversat este:";
echo strrev($sir);
echo strtolower($sir);
echo "<br>";
echo "sirul limbaj se regaseste in sirul $sir de:", "<br>";
echo substr_count($sir, 'limbaj');
```


APLICAȚII PROPUSE

- Testarea celor două exemple date;
- Pentru stringul dat (*$\$sir = \text{"HTML este un limbaj de marcare folosit de browser-ele Web pentru a determina modul de afișare a conținutului paginilor Web"}$*) folosind funcțiile pentru manipularea șirurilor de caractere, să se realizeze următoarele operații:
 - determinarea numărului de cuvinte din șirul de caractere dat;
 - înlocuirea în $\$sir$ a șirului de caractere „HTML”, cu șirul de caractere „HyperText Markup Language”;
 - împărțirea șirului inițial $\$sir$ în mai multe subșiruri, având lungimea de nouă caractere separate prin separatorul “|”;
 - crearea unui vector multidimensional ce va conține părțile din șirul $\$sir$ separate prin separatorul “,”;
 - ștergerea din partea dreaptă a șirului inițial $\$sir$ a caracterelor “folosit de browser-ele Web pentru a determina modul de afișare a conținutului paginilor Web ”;
 - afișarea de două ori a șirului inițial $\$sir$,

APLICAȚII PROPUSE

- amestecarea aleatorie a caracterelor din șirul de caractere $\$s_{ir}$;
- compararea lexicografică a șirului de caractere $\$s_{ir}$ cu șirul de caractere $\$s_{ir1} = \text{“HTML este un limbaj de marcare”}$;
- returnarea poziției de unde începe subșirul “HTML” în șirul de caractere $\$s_{ir}$;
- inversarea șirului de caractere $\$s_{ir}$;
- transformarea tuturor literelor din șirul de caractere $\$s_{ir}$ în litere mici;
- afișarea numărului de apariții ale subșirului “limbaj” în șirul de caractere $\$s_{ir}$.

APLICAȚII PROPUSE

- Exercițiul nr. 3, pag. 101;
- Exercițiile nr. 2 și 3, pag. 102;
- Obs:
 - Pentru aplicațiile nr. 2 și 3, scripturile vor fi definite în cadrul etichetei `<body></body>`;
 - Formularul (*form1.php*) pentru aplicația numărul 3, trebuie să prezinte structura din fig.1;



The screenshot shows a web browser window with the address bar displaying 'localhost/aplicatii/test3.html'. Below the address bar, there are two input fields. The first field is labeled 'a=' and contains the value '0'. The second field is labeled 'b=' and also contains the value '0'. Below these fields, there is a row of seven buttons: 'Adunare', 'Scadere', 'Inmultire', 'Cat impartire', 'Rest impartire', 'Ridicare la putere', and 'resetare'.

Fig.1. Structură *form1.php*

APLICAȚII PROPUSE

- Pentru fiecare operație în parte, scriptul *rezultate.php* pentru aplicația numărul 3 trebuie să prezinte structura prezentată în fig.2.



Fig.2. Structură *rezultate.php*

- Se va utiliza funcția *isset()*, a cărei sintaxă este: *isset (variabila1, variabila2.....);*

-
- **Obs:** se va utiliza cartea “*Programare web. Ghid teoretic și practic*”, autor Șef lucr. dr. mat. Cărbureanu Mădălina, Editura Universității-Petrol Gaze din Ploiești, 2020.

Spor la lucru!