

CAP. 6

ȘIRURI DE CARACTERE (STRINGS)

Un șir de caractere (**string**) este o structură de date ce stochează informație de tip text delimitată de simbolurile " ... " sau ' ... '. PHP prezintă particularități în modul în care sunt folosite șirurile, după cum se poate vedea în exemplele de mai jos.

Observație: Șirurile de caractere sunt expresii (entități ce au și returnează o valoare). Așadar, un string poate fi folosit, pe lângă, afișare, în atribuiri, verificări, etc.

```
$string = "Acesta este un string între ghilimele";  
$stringGol = ""; // Acesta este un string gol
```

Observație: Șirurile de caractere delimitate de " ... " pot conține referințe către alte variabile sau caractere speciale ce vor fi interpretate la afișare.

```
$variabila = "variabila";  
$altaVariabila = 'Aceasta $variabila nu se va afisa';  
print($altaVariabila);  
print("<br>");  
  
$altaVariabila2 = "Aceasta $variabila se va afisa";  
print($altaVariabila2);
```

Operatorul de Concatenare Pentru a concatena două șiruri de caractere se folosește operatorul " . ".

```
$string1 = "Hello World";  
$string2 = "1234";  
echo $string1 . " " . $string2; // rezultat: Hello World 1234
```

În exemplul de mai sus am folosit operatorul " . " de 2 ori pentru a concatena, cele 2 string-uri și un spațiu, rezultatul final fiind: Hello World 1234.

Nu toate caracterele sunt interpretate la fel de PHP. Caracterul "\" poartă denumirea de **escape character** pentru simbolurile cu rol special în PHP. În combinație cu alte caractere acesta îndeplinește funcții speciale. O parte dintre caracterele speciale sunt următoarele:

- \n – este folosit pentru a împinge următorul caracter pe o linie nouă;
- \t – este folosit pentru a reprezenta caracterul "tab";
- \\$ – folosit pentru a reprezenta caracterul "\$";
- \" – folosit pentru a reprezenta caracterul "(";
- \\ – folosit pentru a reprezenta caracterul "\".

```
print "Ma numesc \"Alex\"!"; // sir ce contine ghilimele  
  
print "Text afisat pe 3 linii: \n linia 2 si \n linia 3"; // text  
afisat pe 3 linii (in fisiere text nu in pagini html)  
  
print "Am castigat \$30 :D"; // caracterul dolar are un statut
```

aparte, semnalizand prezenta unei variabile

```
print "Text cu \\backslash\\"; // backslash este un caracter special,  
el poate fi afisat ca atare doar daca este dublat
```

6.1 Funcții predefinite pentru string-uri

Pentru a facilita manipularea string-urilor și a conținutului acestora, PHP pune la dispoziție o serie de funcții predefinite ce pot fi utilizate fără a include alte fișiere sau referințe.

Afișarea lungimii unui string - funcția strlen()

Funcția **strlen()** este folosită pentru a afla lungimea unui șir de caractere. Structura acestei funcții este următoarea:

```
strlen(string $string): int
```

Exemplu de utilizare:

```
echo strlen("Hello world!"); // rezultat: 12
```

Afișarea numărului de cuvinte dintr-un string - funcția str_word_count()

Funcția **str_word_count()** returnează numărul de cuvinte conținute de un string. Structura acestei funcții este următoarea:

```
str_word_count(  
    string $string,  
    int $format = 0,  
    ?string $characters = null  
): array|int
```

Exemplu de utilizare:

```
echo str_word_count("Hello world!"); // rezultat: 2
```

Afișarea caracterelor dintr-un string în ordine inversă - funcția strrev()

Funcția **strrev()** inversează ordinea caracterelor dintr-un string. Structura acestei funcții este următoarea:

```
strrev(string $string): string
```

Exemplu de utilizare:

```
echo strrev("Hello world!"); // rezultat: !dlrow olleH
```

Căutarea poziției unui string într-un alt string - funcția strpos()

Această funcție ne ajută să identificăm dacă un caracter sau un șir de caractere există într-un alt șir de caractere. Dacă acesta există, este returnată poziția sa în string, sau primim un rezultat fals dacă nu se găsește. Structura acestei funcții este următoarea:

```
strpos(string $haystack, string $needle, int $offset = 0): int|false
```

Exemplu de utilizare:

```
$mystring = 'abc';  
$findme   = 'a';  
echo strpos($mystring, $findme);
```

Înlocuirea secvențelor de caractere dintr-un string - funcția `str_replace()`

Această funcție înlocuiește toate secvențele de caractere (căutate) dintr-un string cu altă secvență de caractere. Structura acestei funcții este următoarea:

```
str_replace(  
    array|string $search,  
    array|string $replace,  
    string|array $subject,  
    int &$count = null  
): string|array
```

Exemplu de utilizare:

```
$mystring = 'abc';  
$search    = 'a';  
$replace    = 'b';  
echo str_replace($search, $replace, $mystring);
```

Afișarea unei subsecțiuni de string - funcția `substr()`

Funcția **substr()** returnează o parte din caracterele unui string. Structura acestei funcții este următoarea:

```
substr(string $string, int $offset, ?int $length = null): string
```

Exemplu de utilizare:

```
$sir = "Eu stiu PHP, HTML, CSS";  
echo substr($sir, 0, 7), "<br>"; // rezultat: Eu stiu  
echo substr($sir, 3), "<br>";    // rezultat: stiu PHP, HTML, CSS  
echo substr($sir, -3), "<br>";   // rezultat: CSS  
echo substr($sir, 7, 4), "<br>"; // rezultat: PHP
```

Funcții pentru transformare caracterelor din șirurile de caractere

De multe ori vrem să transformăm caracterele dintr-un string în litere mici / mari sau vrem să avem anumite cuvinte care să înceapă cu literă mare. Pentru a rezolva astfel de probleme putem apela la funcțiile: **strtoupper()**, **strtolower()**, **ucfirst()**, **ucwords()**. Structura tuturor acestor funcții este următoarea:

```
numeFuncție(string $string): string
```

Exemplu de utilizare:

```
$sir = "Eu stiu PHP, HTML, CSS";  
echo strtoupper($sir), "<br>"; // rezultat: EU STIU PHP, HTML,  
CSS  
echo strtolower($sir), "<br>"; // rezultat: eu stiu php, html,
```

```
CSS
echo ucfirst($sir), "<br>";           // rezultat: Eu stiu PHP, HTML,
CSS
echo ucwords($sir), "<br>";           // rezultat: Eu Stiu PHP, HTML,
CSS
```

Eliminarea caracterelor dintr-un string - funcția trim()

Funcția **trim()** elimină spațiile sau orice alte caractere primite ca parametru de la începutul și sfârșitul unui string. Structura acestei funcții este următoarea:

```
trim(string $string, string $characters = " \n\r\t\v\x00"): string
```

Exemplu de utilizare:

```
$str = "Hello World!";
echo $str . "<br>";           // rezultat: Hello World!
echo trim($str, "Hed!");     // rezultat: llo Worl
```

Funcții pentru conversia vectorilor în / din șiruri de caractere

Funcția **explode()** creează un vector dintr-un string folosind ca separator un caracter primit ca parametru. Structura acestei funcții este următoarea:

```
explode(string $separator, string $string, int $limit = PHP_INT_MAX): array
```

Exemplu de utilizare:

```
$pizza = "piecel piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
echo $pieces[0], "<br>"; // rezultat: piecel
echo $pieces[1], "<br>"; // rezultat: piece2
```

Funcția **str_split()** creează un vector dintr-un string împărțind șirul de caractere în subcomponente cu dimensiunea primită ca parametru. Structura acestei funcții este următoarea:

```
str_split(string $string, int $length = 1): array
```

Exemplu de utilizare:

```
$str = "Hello Friend";
$arr1 = str_split($str);
$arr2 = str_split($str, 3);

print_r($arr1); // rezultat: [H, e, l, l, o, , F, r, i, e, n, d]
print_r($arr2); // rezultat: [Hel, lo , Fri, end]
```

Funcția **implode()** creează un string dintr-un vector folosind un caracter primit ca parametru pentru a alipi elementele vectorului. Structura acestei funcții este următoarea:

```
implode(string $separator, array $array): string
```

Exemplu de utilizare:

```
$a = array( 'one', 'two', 'three' );
$b = array( '1st' => 'four', 'five', '3rd' => 'six' );

echo implode( ',', $a ), '<br>'; // rezultat: one,two,three
echo implode( ',', $b );       // rezultat: four,five,six
```

Repetarea unui șir de caractere - funcția str_repeat()

Funcția **str_repeat()** returnează un string repetat de un număr de ori. Structura acestei funcții este următoarea:

```
str_repeat(string $string, int $times): string
```

Exemplu de utilizare:

```
echo str_repeat("-", 10); // rezultat: -----
```

6.2 Alte funcții predefinite pentru string-uri

Pe lângă funcțiile prezentate anterior nucleul PHP mai conține multe altele ce îndeplinesc scopuri bine definite. O listă cu cele mai utilizate funcții destinate manipulării șirurilor de caractere este prezentată în tabelul de mai jos.

Funcție	Descriere
crypt()	Aplică o funcție de hashing
fprintf()	Afișează un string formatat
md5()	Aplică algoritmul MD5 asupra string-ului
sha1()	Aplică algoritmul SHA1 asupra string-ului
str_getcsv()	Transformă un string din format CSV în vector
str_shuffle()	Amestecă aleatoriu caracterele dintr-un string
strip_tags()	Elimină etichetele HTML și PHP dintr-un string
substr_compare()	Compară două string-uri pornind de la o anumită poziție

Teme

1. Realizați o funcție ce primește ca parametru un string. Funcția va elimina toate vocalele din string și va returna string-ul rezultat.
2. Realizați o funcție ce primește ca parametru un string. Aceasta va returna un vector ce reprezintă statistica caracterelor conținute de string-ul primit ca parametru. Ex: pentru "acasă" vectorul rezultat va fi ["a" => 2, "c" => 1, "s" => 1, "ă" => 1].
3. Realizați o funcție ce primește doi parametri (un string și un număr). Funcția va căuta în string-ul primit ca parametru cuvintele cu dimensiunea egală cu numărul primit ca parametru și le va afișa.