Ciągi znaków

Do wyświetlania ciągów znaków na ekranie zwykle używana jest instrukcja ***echo***. Jednak może być używana na kilka sposobów – jest to związane z różnymi sposobami tworzenia napisów. Załóżmy np., że w kodzie mamy trzy zmienne – jedną całkowitą, jedną rzeczywistą i jeden ciąg znaków – utworzone następująco:

$zmienna1=1.7894321;

$zmienna2=154643212;

$zmienna3='abcdEFGH';

Chcielibyśmy teraz wyświetlić je na ekranie wewnątrz innego ciągu znaków w taki sposób, że ciąg będzie przeplatany wartościami zmiennych np. tak:

pierwsza: 1.7894321 druga: 154643212 trzecia: abcdEFGH koniec

To zadanie może być wykonane na kilka sposobów. Najwygodniejsze wydaje się użycie składni ze znakami cudzysłowu:

echo „pierwsza: $zmienna1 druga: $zmienna2 trzecia: $zmienna3 koniec";

Drugi sposób to ręczne składanie wartości całego ciągu za pomocą operatora łączenia łańcuchów.

echo 'pierwsza: '.$zmienna1.' druga: '. $zmienna2.' trzecia: '.$zmienna3.' koniec';

Trzeci sposób to oddzielenie poszczególnych składowych instrukcji echo za pomocą przecinków.

echo 'pierwsza: ', $zmienna1 , ' druga: ' , $zmienna2 , ' trzecia: ' , $zmienna3 , ' koniec';

Czwartym sposobem jest skorzystanie ze składni heredoc. Wystarczy przygotować dodatkową zmienną :

$str=<<<ID

pierwsza: $zmienna1 druga: $zmienna2 trzecia: $zmienna3 koniec

ID;

Oraz użyć jej w instrukcji echo:

echo $str;

Przetwarzanie ciągów znaków

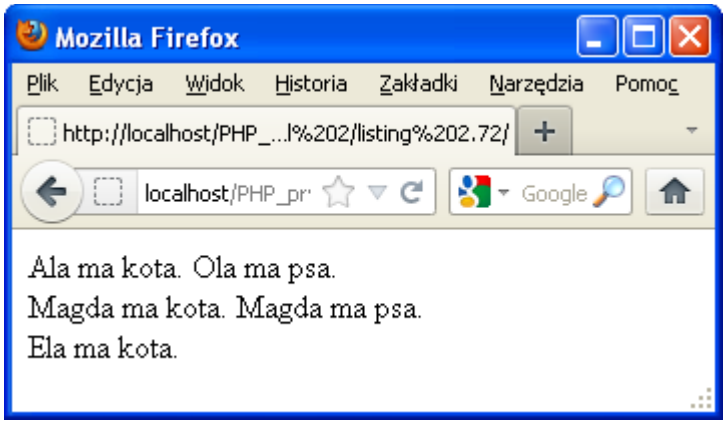
Jeśli chcemy wymienić część łańcucha na inny, możemy zastosować jedną z funkcji zamieniających podciągi ***str_replace, str_ireplace, substr_replace*** lub ***strtr***. Zacznijmy od najczęściej wykorzystywanej funkcji ***str_replace***:

str_replace(str1, str2,str3,&$ile);

Funkcja ta zwraca ciąg ***str3*** przetworzony w taki sposób w którym wszystkie wystąpienia ciągu ***str1*** zostały zmienione na ciąg ***str2***. Jeśli będzie podany opcjonalny parametr ile, zostanie w nim zapisana liczba dokonanych zmian.

Przykład:

```php
<?php
$arr1 = array("%imie1%", "%imie2%");
$arr2 = array("Ala", "Ola");
$arr3 = array("%imie1%", "ma", "kota.");

$str = str_replace($arr1, $arr2, "%imie1% ma kota. %imie2% ma psa.");
echo "$str<br />";

$str = str_replace($arr1, "Magda", "%imie1% ma kota. %imie2% ma psa.");
echo "$str<br />";

$str = str_replace("%imie1%", "Ela", $arr3);
foreach($str as $word) echo("$word ");
?>

```



Odmianą funkcji ***str_replace*** jest ***str_ireplace***. Jej działanie jest analogiczne do ***str_replace***, nie bierze ona jednak pod uwagę wielkości liter.

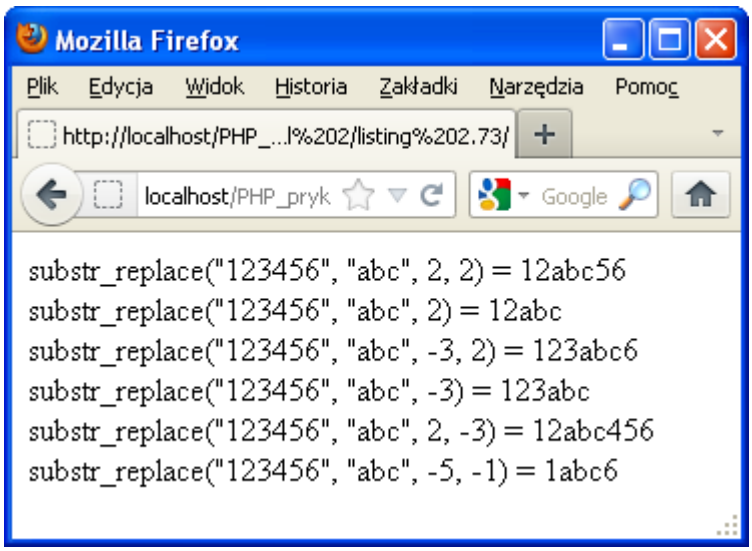
Kolejną funkcją zamieniającą podciągi znaków jest ***substr_replace***

substr_replace(str1,str2, start, ile);

Zwraca ona ciąg ***str1***, z którego począwszy od znaku o indeksie ***start***, zostało wycięte ***ile*** znaków, a w powstałe miejsce został wstawiony ciąg ***str2***.

Przykład:

```php
<?php
echo "substr_replace(\"123456\", \"abc\", 2, 2) = ";
echo substr_replace("123456", "abc", 2, 2);

echo "<br />substr_replace(\"123456\", \"abc\", 2) = ";
echo substr_replace("123456", "abc", 2);

echo "<br />substr_replace(\"123456\", \"abc\", -3, 2) = ";
echo substr_replace("123456", "abc", -3, 2);

echo "<br />substr_replace(\"123456\", \"abc\", -3) = ";
echo substr_replace("123456", "abc", -3);

echo "<br />substr_replace(\"123456\", \"abc\", 2, -3) = ";
echo substr_replace("123456", "abc", 2, -3);

echo "<br />substr_replace(\"123456\", \"abc\", -5, -1) = ";
echo substr_replace("123456", "abc", -5, -1);
?>

```



Kolejna z funkcji to ***strtr***.

strtr(str, znaki1, znaki2)

Zwraca ona ciąg ***str***, w którym znaki zawarte w ***znaki1*** zostały zamienione na odpowiadające im znaki z ciągu ***znaki2***. Jeśli chcielibyśmy na przykład usunąć z dowolnego ciągu wszystkie pisane małymi literami tzw. Polskie znaki i zamienić je na łacińskie odpowiedniki, należałoby zastosować wywołanie:

$str=strtr(„przykładowy ciąg”, „ąćęłńóśźż”, „acelnoszz”);

Porównania

Kolejnymi czynnościami niezbędnymi do sprawnej pracy z PHP są porównania ciągów. Mogą być one wykonywane zarówno za pomocą operatorów porównania, jak i funkcji porównujących. Jedną z takich funkcji jest ***strcmp***, której wywołanie ma schematyczną postać:

strcmp(„ciąg1”, „ciąg2”)

Zwraca ona wartość mniejszą od 0, jeśli ciąg1 jest mniejszy niż ciąg2; większą od 0 jeśli ciąg1 jest większy od ciąg2, lub 0, jeśli ciągi są sobie równe.

Funkcją podobną do ***strcmp*** jest ***strcasecmp***. Działa one w prawie identyczny sposób, lecz nie bierze pod uwagę wielkości liter.

Obie funkcje mają również swoje odmiany, które pozwalają na zdefiniowanie, ile znaków z pierwszego i drugiego ciągu będzie podlegało porównaniu. Są to: ***strncmp*** i ***strncasecmp***. Wywołanie ***strncmp*** (i analogicznie ***strncasecmp***) ma schematyczną postać:

strncmp(„ciąg1”, „ciąg2”, ile)

Przeszukiwanie

Przeszukiwanie ciągów to kolejna przydatna operacja. Na jej wykonywanie pozwala cały zestaw funkcji są to m.in.: ***strpos, strrpos, stripos, strripos, strstr, stristr***. Funkcja ***strpos*** zwraca numer pierwszej pozycji poszukiwanego ciągu. Jej wywołanie ma postać:

strpos(„ciąg przeszukiwany”, „ciąg poszukiwany”, start)

Przeszukiwanie rozpoczyna się od początku ciągu lub od pozycji wskazywanej przez opcjonalny parametr start. Przy tym pozycje ciągu przeszukiwanego numerowane są od 0. Przykładowe wywołanie:

$pos=strpos(„abcdefg”, „cde”);

Spowoduje przypisanie zmiennej ***$pos*** wartości 2.

Odmianą ***strpos*** jest ***stripos***. Działa ona prawie identycznie jak ***strpos***, lecz nie rozróżnia wielkości liter. Jeżeli chcemy przeszukiwać ciągi od końca, czyli uzyskać pozycję ostatniego

wystąpienia poszukiwanego ciągu, należy skorzystać z funkcji ***strrpos*** lub ***strripos***. Przykładowe wywołanie:

$pos=strrpos(„abcdabcd", „ab");

Spowoduje przypisanie zmiennej ***pos*** wartości 4.

Jeżeli w przeszukiwanym ciągu nie ma poszukiwanego, każda z wymienionych funkcji zwraca wartość ***false***.

Oprócz opisanych wyżej istnieją jeszcze dwie inne przydatne funkcje przeszukujące ciągi – ***strstr*** i ***stristr***. Jak łatwo się domyślić, ***stristr*** to odmiana funkcji ***strstr***, która nie uwzględnia wielkości znaków. Wywołanie ***strstr*** ma postać:

strstr(„ciąg przeszukiwany", „ciąg poszukiwany", przed)

Zwraca ona część przeszukiwanego ciągu, która rozpoczyna się od ciągu poszukiwanego. Przykładowe wywołanie:

$str=strstr(„abcdefg", „de");

Spowoduje przypisanie zmiennej ***$str*** wartości ***defg***.

Jeśli zostanie podany trzeci opcjonalny argument i będzie miał wartość ***true***, zwrócona zostanie część ciągu przeszukiwanego, która znajduje się przed ciągiem przeszukiwanym. Przykładowe wywołanie:

$str=strstr(„abcdefg", „de", true);

Spowoduje zatem przypisanie zmiennej ***$str*** wartości ***abc***.

Ćwiczenia do samodzielnego wykonania

Ćwiczenie 1.

Napisz funkcję, która będzie przetwarzała ciąg znaków w taki sposób, że wszystkie małe litery zostaną zamienione na wielkie a wielkie na małe.

Ćwiczenie 2.

Napisz funkcję wykonującą cenzurowanie tekstu. Powinna ona przyjmować dwa argumenty, z których pierwszym byłby ciąg podlegający cenzurowaniu, a drugim tablica zawierająca niepożądane zwroty. Każdy zwrot znajdujący się w tablicy powinien zostać zamieniony w tekście na ciąg [ocenzurowano].

Ćwiczenie 3.

Napisać funkcję, która będzie obliczała liczbę wystąpień określonego ciągu znaków w danym tekście. Tekst i szukany ciąg znaków powinny być przekazywane w postaci argumentów.