

## Osnove PHP-a (5)

### Promena tipa PHP promenljive

Kao što smo spomenili na početku ovog poglavlja, PHP podržava veliki broj različitih tipova promenljivih (celobrojne tj integer, zatim float, logičke boolean, nizovne array, object, resource i nizovni tim string). Detaljnije ovim tipovima ćemo se pozabaviti u sledećem poglavlju pod nazivom Razumevanje tipova PHP promenljivih. Prvo ćemo da razmotrimo mogućnost promena tipa promenljive pošto je ona stvorena.

PHP je *“loosely typed”* jezik, za razliku programerskih jezika kao što je Java koji je *“strongly typed”* jezik. Pravila, *“strongly typed”* su takva da, jednom pošto se deklarira promenljiva kao neki određeni tip, taj tip se kasnije ne može menjati. U Javi, na primer, se ne može dodeliti tip folat promenljivoj koja je inicijalno definisana kao nizovni tip tj string tip.

```
<?php
$myNumber = 6; // variable is of integer type

$myNumber = six; // variable has now changed to string type

?>
```

“Loosely typed” jezik kao što je PHP (i JavaScript), sa druge strane, dozvoljava promenu tipa promenljive, u bilo trenutku života promenljive, jednostavno dodeljivanjem vrednosti drugog tipa toj promenljivoj. Na primer, možemo stvoriti promenljivu celobrojnog tipa , i kasnije joj promeniti tip u nizovni tip bez ikakvog problema. Proces dinamičke promene tipa promenljive je poznat kao proces automatske konverzije tipa promenljive.

### Provera da li je promenljivoj dodeljena vrednost

Kada radimo sa promenljivim, često je neophodno da se uverimo da li je promenljivoj dodeljena vrednost koja je povezana ili nije sa nekim delom skripta. Mehanizam koji obezbeđuje proveru ove sposobnosti, se zadaje preko funkcije *isset()*. Da bismo testirali da li promenljiva ima neku vrednost jednostavno pozivamo funkciju *isset()* koristeći ime promenljive kao argument. Sledeći primer koda testira proveru da li promenljiva ima neku vrednost ili ne i ispisuje poruku shodno tome:

```
<?php
$myVariable = "hello";

if (isset($myVariable))
{
    echo "It is set.";
}
else
{
    echo "It is not set.";
}
?>
```

## Razumevanje tipova PHP promenljivih

U ovom poglavlju, razmatraćemo PHP promenljive tipa *integer*, *string*, *float* i *Boolean*. Pošto su promenljive tipa *array* i *object* malo kompleksniji entiteti, njih ćemo razmatrati u naknadnim poglavljima u knjizi. (Pogledati poglavlje Korišćenje nizova u PHP-u i PHP objektno-orjentisano programiranje).

### PHP promenljiva tipa integer

Promenljive tipa integer su u mogućnosti da koriste ceo raspon borjeva od -2147483648 do 2147483647. Negativne vrednosti mogu biti dodeljene, tako što se stavlja znak minus (-) posle dodeljenog operatora i ispred broja. U slučaju da se vrednost dodeljena promenljivoj tipa *integer* nalazi van dozvoljenog opsega, dobijena dodelom ili kao rezultat neke matematičke operacije, tip promenljive se automatski konvertuje i promenljivu tipa *float*.

Sledeći primer pokazuje dodeljivanje vrednosti promenljivima tipa integer.

```
<?php
$myInteger = 10;
$myNegative = -13457231;
?>
```

Vrednosti promenljive *integer* se mogu ispisati u oktalnom sistemu dodavanjem nule "0" kao prefiksa.

```
$myOctInteger = 0456;
```

Isto tako, heksadecimalnim vrednostima se kao prefiks stavlja 0x.

```
$myHexInteger = 0x5EF3;
```

### PHP promenljiva tipa float

Promenljive tipa float sadrže brojeve koji zahtevaju ispis preko decimalnog zapisa., i one mogu mnogo veći raspon brojeva u odnosu na raspon integer promenljivih. (Kao što su 1.067, 0.25, 423454567098, 84664435.9576).

Stvaranje i inicijalizacija promenljivih tipa float je sledeća:

```
<?php
$myFloat = 9234.98;
$myOtherFloat = 9547894367.987483701
?>
```

## PHP promenljiva tipa boolean

PHP promenljiva tipa Boolean može imati vrednost *true* ili *false*, i koriste se za testiranje uslova da li neki deo script-a radi ispravno ili ne. Detaljnije ćemo se pozabaviti promenljivima tipa Boolean u poglavlju PHP Flow kontrola i Looping, sa posebnim osvrtom na if komandu. Korisno je znati da se vrednosti *true* i *false* u PHP-u predstavljaju kao brojevi 1 i 0.

## PHP promenljiva tipa string

Promenljive tipa string se koriste kako bi grupisale alfanumerički niz znakova kao što su reči i rečenice. Pored mehanizma za kreiranje i promenu celog stringa vrednosti promenljivih, PHP dopušta da se uzimaju i menjaju delovi stringa.

String se može dodeliti promenljivoj ili stavljanjem niza znakova između jednostrukih navodnika ili dvostrukih navodnika. Ako je u okviru niza već u upotrebi neka vrsta navodnika, jednostruki ili dvostruki, potrebno je koristiti suprotni kako bi se obuhvatio niz.

```
<?php
$string = "A string of text";
$string2 = 'Another string of text';
$string3 = "This string contains 'single quotes'";
$string4 = 'This string contains "double quotes"';
?>
```

Takođe, može se izbeći korišćenje navodnika u stringu, pristupajući im preko znaka (\).Ovaj način je pre svega preporučljiv za stringove koji koriste istovremeno jednostruke i dvostruke navodnike, kako ne bi došlo do zbunjivanja pretprocesora.

```
<?php
$string3 = 'This string contains \'single quotes\'';
$string4 = "This string contains \"double quotes\" and \'single quotes\'';
?>
```

Dvostruki navodnici takođe dozvoljavaju ubacivanje specijalnih kontrolnih rečenica, koje imaju specijalno značenje za PHP pretprocesor (kao što su znak "tab" ili novi red). Sledeća tabela prikazuje različite kontrole sekvenci i njihov opis.

Kontrolne sekvence	Opis
\n	Novi red
\r	Carriage Return
\t	Tab
\\	Backslash karakter
\"	Dvostruki navodnici
\\$	Dollar znak (mera preventive kako se tekst ne bi tretirao kao ime promenljive)
\034	Octal ASCII vrednost
\x0C	Hexadecimal ASCII vrednost

Kao primer, možemo deklarirati promenljivu tipa string koja sadrži znak tab i novi red, na sledeći način:

```
<?php
$string = "This is a line of Text \n and this is another line with a tab here
\t for us.";
?>
```

## Vađenje i pisanje delova stringa

Jednom pošto smo definisali promenljivu tipa string, možemo izvaditi ili načiniti promene pojedinih karaktera u stringu korišćenjem izraza {x}, gde x predstavlja index karaktera u stringu koji želimo da vidimo ili promenimo. Pre nego što pogledamo primer, važno je da imamo na umu da numeracija karaktera u stringu počinje od broja 0. Znači prvi karakter u stringu ima poziciju 0, a ne poziciju 1. Na primer, ovde ćemo izvršiti promenu prvog i poslednjeg karaktera u sledećem stringu:

```
<?php
$string = "My Bug";
$string{0} = "m";
$string{5} = "s";
echo $string;
?>
```

Gore napisani script, će izvršiti promenu stringa My Bug u my Bus.

## Kreiranje PHP heredoc stringova

PHP heredoc string sintaksa dozvoljava upotrebu free form teksta bez ikakvih brriga oko upotrebe specijalnih karaktera kao što su navodnici i back slashes. Sadržaj heredoc stringa je obuhvaćen sa <<<EOD i EOD; markerima.

Jedino pravilo je to, da EOD koje se koristi za završnicu, mora biti na početku poslednjeg reda. Takođe EOD mora biti jedini sadržaj tog poslednjeg reda, što znači da ništa ne sme u njemu da se nalazi osim makrera EOD, što se vidi na sledećem primru:

```
<?php

$string = <<<EOD
This is some free form text. It can span multiple lines and can contain otherwise
troublesome characters like \ and " and ' without causing any problems.
EOD;
echo $string;

?>
```