**PHP ლექცია №6. ტიპების არაცხადი და ცხადი გამოცხადება. ტიპების შეთანხმება.**

მინიჭების ოპერაციით, თუ კი, ვთვათ, შესაძლებელია ცვლადების ინიციალიზაცია - შექმნა, ასევე შესაძლებელია მათი ანულირება - გაუქმება. ამისთვის გამოიყენება კონსტრუქ­ცია unset() , რომელსაც აქვს შემდეგი სინტაქსისი:

**unset($var[,var1[,….]])**

unset() კონსტრუქციის მრგვალ ფრჩილებში მოთავსებულია ერთი ან მძიმეებით გამოყენებუ­ლი რამდენიმე ცვლადი, რომლის გაუქმებაა საჭირო. unset() კონსტრუქციის გამოძახების შემ­დეგ, მეხსიერება, რომელიც გამოყოფილია აღნიშნული ცვლადებისათვის, უბრუნდება სის­ტემას, ხოლო თავად ამ ცვლადებს ექნებათ მნიშვნელობა NULL. unset() კონსტრუქცია გამო­იყენება ყველა ტიპის ცვლადებისთვის.

ლისტინგი 1. ცვლადების გაუქმება unset()

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

<?php

$str1="ჯერ შევქმენით ცვლადები: ";

$str2="მთელი";

$str3=12;

$str4="ნამდვილი";

$str5=25.6;

echo "$str1 ' $str2=' $str3 ' $str4= ' $str5,<br>" ;

unset($str1,$str2,$str3,$str4, $str5);

print "ცვლადების დაცლის შემდეგ";

echo "<br>$str1 ' $str2=' $str3 '$str4=' $str5<br>" ;

var\_dump($str1,$str2,$str3,$str4,$str5);

?>

</BODY>

</HTML>

ლისტინგი 1-ის გაშვების შემდეგ მივიღებთ

შედეგი 1

**ჯერ შევქმენით ცვლადები: ' მთელი=' 12 ' ნამდვილი= ' 25.6
ცვლადების დაცლის შემდეგ
' =' '='
NULL NULL NULL NULL NULL**

როგორც გვახსოვს, სტრინგისთვის - ტექსტური ცვლადისთვის მეხსიერების შესაძლებელი მაქსიმალური მნიშვნელობა, მისთვის გამოყოფილი მეხსიერებაა; ამიტომაც, ცვლადების ანუ­ლირებით შეგვიძლია დავზოგოთ მეხსიერება და ანულირება მოვახდინოთ უკვე არა საჭირო ცვლადებისა.

რადგანაც სკრიპტში შესაძლებელია იყოს არაინიციალიზირებული ან გაუქმებული ცვლადი, ამიტომ საჭიროა გვქონდეს შესაძლებლობა შევამოწმოთ, არსებობს თუ არა, ესა თუ ის ცვლა­დი, ან ცვლადები. ჩვენ უკვე გავეცანით ცვლადების ტიპის ამომცნობ ფუნქციას **var\_dump()** -ს ეხლა გავეცნოთ სხვა საშუალებებს, კერძოს isset()-ს, რომელსაც გააჩნია შემდეგი სინტაქსისი:

**isset($var[,var1[,….]])**

**isset** () კონსტრუქციის მრგვალ ფრჩილებში მოთავსებულია ერთი ან მძიმეებით გამოყენებუ­ლი რამდენიმე ცვლადი. თუ კი ყველა ეს ცვლადები არსებობენ, კონსტრუქცია აბრუნებს **TRUE** -ს, ხოლო თუ კი ერთი ცვლადიც მაინც არ არსებობს, მაშინ ბრუნდება **FALSE** -ს.

ლისტინგი 2. ცვლადების შემოწმება isset()

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 // გამოვაცხადოთ ცარიელი ცვლადი

 $str = '';

 if(isset($str)) // isset აბრუნებს მნიშვნელობა TRUE-ს, ამოტომ სრულდება მომდევნო ოპერატორი

 echo '1. ცვლადი $str არსებობს <br>';

 // $str -ს არაინიციალიზაციას ვახდენთ

 $str = NULL;

 if(isset($str)) // isset აბრუნებს მნიშვნელობა FALSE-ს, ამოტომ არ სრულდება მომდევნო ოპერატორი

 echo '2. ცვლადი $str ისევ არსებობს <br>';

 // ვახდენთ $a და $b ცვლადების ინიციალიზირებას

 $a = "variable";

 $b = "another variable";

 // შევამოწმოთ ცვლადების არსებობა

 if(isset($a)) // TRUE

 echo '3. ცვლადი $a არსებობს<br>';

 if(isset($a, $b)) // TRUE

 echo '4. ცვლადები $a და $b არსებობენ<br>';

 // მოვახდინოთ $a ცვლადის ანულირება

 unset ($a);

// ვამოწმებთ ცვლადების არსებობას

 if(isset($a)) // FALSE

 echo '5. ცვლადი $a არსებობს<br>';

 if(isset($a, $b)) // FALSE

 echo '6. ცვლადები $a და $b არსებობენ<br>';

?>

</BODY>

</HTML>

ლისტინგ 2-ის შესრულების შემდეგ

შედეგი 2.

**1. ცვლადი $str არსებობს
3. ცვლადი $a არსებობს
4. ცვლადები $a და $b არსებობენ**

როგორც ვხედავთ, ცარიელი სტრიქონი არ არის ექვივალენტური არაინიცირებული ცვლა­დის (იმ ცვლადისა, რომელიც იღებს მნიშვნელობას). იმის შესამოწმებლად, არის თუ არა სტრიქონი ცარიელი თუ არა, გამოიყენება სპეციალური კონსტრუქცია - empty(), რომელსაც შემდეგი სინტაქსისი აქვს:

**empty($var)**

empty() კონსტრუქციაში isset()-სგან განსხვავებით ერთი ცვლადი მოიცემა მრგვალ ფრჩხი­ლებში. empty() კონსტრუქცია აბრუნებს **TRUE** -ს, როცა ცვლადი ცარიელია ””, ნულის ტო­ლია, ტექსტურ ცვლადში არის მხოლოდ ნული - ”0”, NULL, FALSE, ცარიელი მასივი, არაინი­ცირებული ცვლადი (NULL-ის ანალოგიური შემთხვევა). ყველა სხვა შემთხვევაში ბრუნდება **FALSE**.

ლისტინგი 3. ცარიელი ცვლადის შემოწმება empty()

 <!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 // ვაცხადებთ ცარიელ ცვლადს

 $str = '';

 // ვამოწმებთ ცვლადის არსებობას

 if(isset($str)) // TRUE

 echo '$str ცვლადი არსებობს<br>';

 // ვამოწმებთ ხომ არაა ცვლადი ცარიელი

 if(empty($str)) // TRUE

 echo '$str ცვლდი ცარიელია<br>';

?>

</BODY>

</HTML>

ლისტინგ 3-ის შესრულების შემდეგ

შედეგი 3

**$str ცვლადი არსებობს
$str ცვლდი ცარიელია**

**in\_array**

in\_array — ამოწმებს, არსებობს თუ არა მასივში ელემენტი-მნიშვნელობა

**სინტაქსისი**

bool **in\_array** ($needle , array $haystack [, bool $strict = **FALSE** ] )

ხდება *haystack* მასივში *needle*-ის მნიშვნელობა. თუ *strict* не მითითებულია, მაშინ ძებნისას გამოყენებული იქნება არამკაცრი შედარება (”==”).

პარამეტრების სია

*needle* - საძებნი მნიშვნელობა. თუ *needle* - სტრიქონია, მაშინ შედარება ხდება რეგისტრის გათვალისწინებით.

*haystack* - მასივია, სადაც ხდება მნიშვნელობის ძებნა.

*strict* - თუ კი ის მითითებულია *strict* -ის მნიშვნელობა **TRUE** , მაშინ შედარება ხდება ტიპების მიხედვით, ანუ მკაცრი შედარებით (”===”).

აბრუნებს **TRUE** -ს, თუ კი მასივში საძებნი ელემენტი არსებობს, წინააღმდეგ შემთხვევაში **FALSE**-ს.

შემდეგი ფუნქციები, ძირითადად ცვლადის ტიპიდან გამომდინარე, აბრუნებენ ამა თუ იმ მნიშვნელობებს; ისინი ხშირად გამოიყენებიან პირობით ოპერატორში

ცხრილი 1

|  |  |
| --- | --- |
| is\_array($var) | აბრუნებს TRUE -ს, თუ $var არის მასივი, წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_bool($var) | აბრუნებს TRUE -ს, თუ $var არის ლოგიკური ტიპის ცვლადი, წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_callable($var) | აბრუნებს TRUE -ს, თუ $var არის ფუნქცია, წინაამდეგ შემთხვევაში FALSE -ს საქმე იმა­შია, რომ რიგი ფუნქციები პარამეტრად ღებულობენ სამომხმარებლო ფუნქციების სახელებს, და ამ ფუნქციებზე ხდება მიმარ­თვა მათი შესრულებისას. ფუნქცია is\_callable() ამოწმებს, შეესაბამება თუ არა $var ცვლადს რომელიმე არსებული ფუნქცია |
| is\_double($var) | აბრუნებს TRUE -ს, თუ $var არის double (ან float) ტიპის ცვლადი, წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_float($var) | is\_double() ფუნქციის სინონიმი |
| is\_int($var) | აბრუნებს TRUE -ს, თუ $var არის მთელი ტიპის ცვლადი int, წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_integer($var) | is\_int() ფუნქციის სინონიმი |
| is\_long($var) | is\_int() ფუნქციის სინონიმი |
| is\_null($var) | აბრუნებს TRUE -ს, თუ $var NULL -ის ტოლია, წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_numeric($var) | აბრუნებს TRUE -ს, თუ $var არის რიცხვითი (int, float ან double) ტიპის ცვლადი ან რიცხვის შემცველი სტრიქონული ტიპის ცვლადი, წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_object($var) | აბრუნებს TRUE -ს, თუ $var არის ობიექტი , წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_real($var) | is\_float() ფუნქციის სინონიმი |
| is\_resource($var) | აბრუნებს TRUE -ს, თუ $var არის დისკრიპ­ტორი , წინაამდეგ შემთხვევაში FALSE -ს  |
| is\_scalar($var) | აბრუნებს TRUE -ს, თუ $var არის სკალჲარი (int, float, double, string или boolean), წინა­აღმდეგ შემთხვევაში (array, object ან resource) FALSE -ს  |
| is\_string($var) | აბრუნებს TRUE -ს, თუ $var არის ტექსტური, სტრინგული ტიპის ცვლადი, წინაამდეგ შემ­თხვევაში FALSE -ს  |

ლისტინგი 4. ცვლადთა ტიპების მიხედვით მნიშვნელობის დაბრუნება

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 // გამოვაცხადეთ ნამდვილ ცვლადი

 $float = 123.24;

 if(is\_numeric($float)) // TRUE

 echo '$float ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ მთელ ცვლადი

 $number = 0;

 if(is\_numeric($number)) // TRUE

 echo '$number ცვლადი არის რიცხვითი<br>';

 /\* გამოვაცხადეთ სტრიქონული ცვლადი,

 რომელიც შეიცავს რიცხვს \*/

 $str = '12';

 if(is\_numeric($str)) // TRUE

 echo '$str ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ ცარიელი სტრიქონული ცვლადი

 $emptystr = '';

 if(is\_numeric($emptystr)) // FALSE

 echo '$emptystr ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ ცარიელი სტრიქონული ცვლადი,

 // შეიცავს არასწორ რიცხვს

 $wrongstr = '12q';

 if(is\_numeric($wrongstr)) // FALSE

 echo '$wrongstr ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ ლოგიკური ცვლადი,

 $flag = TRUE;

 if(is\_numeric($flag)) // FALSE

 echo '$flag ცვლადი არის რიცხვითი<br>';

?>

</BODY>

</HTML>

ლისტინგი 4-ის შესრულების შედეგად გვაქვს

$float ცვლადი არის რიცხვითი
$number ცვლადი არის რიცხვითი
$str ცვლადი არის რიცხვითი

get\_defined\_vars() აბრუნებს მრავალგანზომილებიან მასივში განსაზღვრულ ცვლადებს, იქ­ნება ეს გარემოს ცვლადებს, სერვერულ ცვლადებს, ცვლადებს, რომელსაც ქმნის მომხმარე­ბელი

ლისტინგი 5 get\_defined\_vars()

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 // გამოვაცხადეთ ნამდვილ ცვლადი

 $float = 123.24;

 if(is\_numeric($float)) // TRUE

 echo '$float ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ მთელ ცვლადი

 $number = 0;

 if(is\_numeric($number)) // TRUE

 echo '$number ცვლადი არის რიცხვითი<br>';

 /\* გამოვაცხადეთ სტრიქონული ცვლადი,

 რომელიც შეიცავს რიცხვს \*/

 $str = '12';

 if(is\_numeric($str)) // TRUE

 echo '$str ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ ცარიელი სტრიქონული ცვლადი

 $emptystr = '';

 if(is\_numeric($emptystr)) // FALSE

 echo '$emptystr ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ ცარიელი სტრიქონული ცვლადი,

 // შეიცავს არასწორ რიცხვს

 $wrongstr = '12q';

 if(is\_numeric($wrongstr)) // FALSE

 echo '$wrongstr ცვლადი არის რიცხვითი<br>';

 // გამოვაცხადეთ ლოგიკური ცვლადი,

 $flag = TRUE;

 if(is\_numeric($flag)) // FALSE

 echo '$flag ცვლადი არის რიცხვითი<br>';

 **$arr=get\_defined\_vars();**

 **echo "გამოგვაქვს ყველა გამოცხადებული ცვლადი $arr";**

 **var\_dump($arr);**

?>

</BODY>

</HTML>

ლისტინგი 5-ის შესრულების შედეგად გვაქვს

$float ცვლადი არის რიცხვითი
$number ცვლადი არის რიცხვითი
$str ცვლადი არის რიცხვითი
გამოგვაქვს ყველა გამოცხადებული ცვლადი Arrayarray(20) { ["GLOBALS"]=> &array(21) { ["GLOBALS"]=> \*RECURSION\* ["\_ENV"]=> array(33) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" } ["HTTP\_ENV\_VARS"]=> array(33) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" } ["\_POST"]=> array(0) { } ["HTTP\_POST\_VARS"]=> array(0) { } ["\_GET"]=> array(0) { } ["HTTP\_GET\_VARS"]=> array(0) { } ["\_COOKIE"]=> array(2) { ["\_ga"]=> string(25) "GA1.2.99260255.1413927363" ["\_gat"]=> string(1) "1" } ["HTTP\_COOKIE\_VARS"]=> array(2) { ["\_ga"]=> string(25) "GA1.2.99260255.1413927363" ["\_gat"]=> string(1) "1" } ["\_SERVER"]=> array(35) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["PHP\_SELF"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REQUEST\_TIME"]=> int(1413927576) } ["HTTP\_SERVER\_VARS"]=> array(35) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["PHP\_SELF"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REQUEST\_TIME"]=> int(1413927576) } ["\_FILES"]=> array(0) { } ["HTTP\_POST\_FILES"]=> array(0) { } ["\_REQUEST"]=> array(2) { ["\_ga"]=> string(25) "GA1.2.99260255.1413927363" ["\_gat"]=> string(1) "1" } ["float"]=> float(123.24) ["number"]=> int(0) ["str"]=> string(2) "12" ["emptystr"]=> string(0) "" ["wrongstr"]=> string(3) "12q" ["flag"]=> bool(true) ["arr"]=> \*RECURSION\* } ["\_ENV"]=> array(33) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" } ["HTTP\_ENV\_VARS"]=> array(33) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" } ["\_POST"]=> array(0) { } ["HTTP\_POST\_VARS"]=> array(0) { } ["\_GET"]=> array(0) { } ["HTTP\_GET\_VARS"]=> array(0) { } ["\_COOKIE"]=> array(2) { ["\_ga"]=> string(25) "GA1.2.99260255.1413927363" ["\_gat"]=> string(1) "1" } ["HTTP\_COOKIE\_VARS"]=> array(2) { ["\_ga"]=> string(25) "GA1.2.99260255.1413927363" ["\_gat"]=> string(1) "1" } ["\_SERVER"]=> array(35) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["PHP\_SELF"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REQUEST\_TIME"]=> int(1413927576) } ["HTTP\_SERVER\_VARS"]=> array(35) { ["PHP\_FCGI\_CHILDREN"]=> string(1) "0" ["PATH"]=> string(29) "/bin:/usr/bin:/sbin:/usr/sbin" ["PWD"]=> string(8) "/opt/php" ["SHLVL"]=> string(1) "0" ["PHP\_FCGI\_MAX\_REQUESTS"]=> string(1) "0" ["FCGI\_ROLE"]=> string(9) "RESPONDER" ["UNIQUE\_ID"]=> string(24) "VEbSmB@qpCIAAEiVFCkAAACD" ["BINARY\_DOCROOT"]=> string(28) "/home/u392729507/public\_html" ["BINARY\_SERVERADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["HTTP\_REFERER"]=> string(51) "http://kereseli.besaba.com/PHP/laboratoriuli\_6.html" ["HTTP\_HOST"]=> string(19) "kereseli.besaba.com" ["HTTP\_COOKIE"]=> string(37) "\_ga=GA1.2.99260255.1413927363; \_gat=1" ["HTTP\_X\_REAL\_IP"]=> string(12) "95.104.9.188" ["HTTP\_CONNECTION"]=> string(5) "close" ["HTTP\_USER\_AGENT"]=> string(65) "Mozilla/5.0 (Windows NT 5.1; rv:32.0) Gecko/20100101 Firefox/32.0" ["HTTP\_ACCEPT"]=> string(63) "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8" ["HTTP\_ACCEPT\_LANGUAGE"]=> string(14) "en-US,en;q=0.5" ["SERVER\_SIGNATURE"]=> string(0) "" ["SERVER\_SOFTWARE"]=> string(6) "Apache" ["SERVER\_NAME"]=> string(19) "kereseli.besaba.com" ["SERVER\_ADDR"]=> string(14) "31.170.165.178" ["SERVER\_PORT"]=> string(2) "80" ["REMOTE\_ADDR"]=> string(12) "95.104.9.188" ["DOCUMENT\_ROOT"]=> string(28) "/home/u392729507/public\_html" ["SERVER\_ADMIN"]=> string(29) "webmaster@kereseli.besaba.com" ["SCRIPT\_FILENAME"]=> string(78) "/home/u392729507/public\_html/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REMOTE\_PORT"]=> string(5) "48669" ["GATEWAY\_INTERFACE"]=> string(7) "CGI/1.1" ["SERVER\_PROTOCOL"]=> string(8) "HTTP/1.1" ["REQUEST\_METHOD"]=> string(3) "GET" ["QUERY\_STRING"]=> string(0) "" ["REQUEST\_URI"]=> string(41) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/" ["SCRIPT\_NAME"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["PHP\_SELF"]=> string(50) "/PHP/laboratoriuli\_6/lab\_6\_5\_get\_defined/index.php" ["REQUEST\_TIME"]=> int(1413927576) } ["\_FILES"]=> array(0) { } ["HTTP\_POST\_FILES"]=> array(0) { } ["\_REQUEST"]=> array(2) { ["\_ga"]=> string(25) "GA1.2.99260255.1413927363" ["\_gat"]=> string(1) "1" } ["float"]=> float(123.24) ["number"]=> int(0) ["str"]=> string(2) "12" ["emptystr"]=> string(0) "" ["wrongstr"]=> string(3) "12q" ["flag"]=> bool(true) }

**gettype($var)** ახდენს ცვლადის ტიპის განსაზღვრას, აბრუნებს ცვლადის ტიპს, რომელსაც შეიძლება ჰქონდეს ერთ-ერთი შემდეგი მნიშვნელობა: "boolean", "integer", "double" (double და float -ისთვის), "string", "array", "object", "resource", "NULL" და "unknown type".

ლისტინგი 6 ტიპის განსაზღვრა gettype-ით

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 // გამოვაცხადეთ მთელ ცვლადი

 $number = 123;

 echo gettype($number); // integer ცვლადები და მონაცემთა ტიპები

 echo '<br>';

 // გამოვაცხადეთ ლოგიკური ცვლადი,

 $flag = TRUE;

 echo gettype($flag); // boolean

 echo '<br>';

 // გამოვაცხადეთ სტრიქონული ცვლადი

 $str = '';

 echo gettype($str); // string

 echo '<br>';

 // გამოვაცხადეთ ნამდვილ ცვლადი

 $var = 1.7;

 echo gettype($var); // double

 echo '<br>';

 // ვანადგურებთ ცვლადს $var

 unset($var);

 echo gettype($var); // NULL

 echo '<br>';

?>

</BODY>

</HTML>

ლისტინგ 6-ის შესრულებით მივიღებთ

integer
boolean
string
double
NULL

 is\_resource() ფუნქციას შეუძლია განსაზღვროს - არის თუ არა ცვლადი დისკრიპტორი. ეს ფუნქცია ხშირად გამოიყენება get\_resource\_type() -ფუნქციასთან ერთად, რომელიც აბრუნებს დისკრიპტორის ტიპს. ლისტინგ 7-ში მოყვანილია მაგალითი, რომელიც ახდენს შეერთებას MySQL მონაცემთა ბაზების მართვის სისტემასთან mysql\_connect() ფუნქციის საშუალებით, რომელიც აბრუნებს შეერთების დესკრიპტორს, როცა შეერთება მოხდა წარმატებით, აბრუნებს FALSE -ს წინააღმდეგ შემთხვევაში.

ლისტინგი 7 დესკრიპტორი is\_resource() და get\_resource\_type()

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 // ვახდენთ დაკავშირებას MySQL -თან

 $dbcnx = mysql\_connect("localhost", "root", "");

 // ვამოწმებთ არის თუ არა ცვლადი $dbcnx

 // შეერთების დესკრიპტორი

 if(is\_resource($dbcnx))

 {

 // ცვლადი $dbcnx არის დესკრიპტორი.

 // გამოგვყავსდესკრიპტორის ტიპი ბრაუზერის ფანჯარაში

 echo get\_resource\_type($dbcnx); // mysql link

 }

?>

</BODY>

</HTML>

ლისტინგ 7-ის შესრულებისას, თუ კი mysql\_connect() ფუნქცია წარმატებით ამყარებს შეერ­თებას მონაცემთა ბაზების მართვის სისტემა (მბმს) MySQL -თან, მაშინ ცვლადი $dbcnx ხდება დესკრიპტორი ( წინააღმდეგ შემთხვევაში იღებს მნიშვნელობას FALSE -ს და არის ლოგიკური ტიპის). თუ კი ეხლა ამ დესკრიპტორს გადავცემთ get\_resource\_type() ფუნქციას, მაშინ იგი აბ­რუნებს მის ტიპს სტრიქონით "mysql link".

**ტიპების არაცხადი დაყვანა** - სუსტად ტიპიზირებულ ენებში, ისეთში როგორიცაა PHP, არაა საჭირო წინასწარ, ცხადი სახით ცვლადის ტიპის მითითება. დაპროგრამირების ასეთი ტიპის ენაში , იმის და მიხედვით, თუ რა მნიშვნელობა მიენიჭა ცვლადს, ავტომატურად განისაზ­ღვრება მისი ტიპი.გარდა ამისა ასეთი ცვლადების ტიპები იცვლება ხოლმე კონტექსტიდან გამომდინარე - მაგალითად თუ კი ეს ცვლადი გამოიძახება რომელიღაც გამოსახულებაში, რომელშიც უკვე მონაწილეობს და ჩართულია სხვა ტიპის ცვლადი, მაშინ PHP შეეცდება ავ­ტომატურად მიუსადაგოს, მიიყვანოს ახლადგამოძახებული ცვლადის ტიპი საჭირო ტიპა­მდე. კონკრეტულად, თუ კი სტრიქონული, ტექსტური ცვლადი შეიცავს რიცხვს და ის გა­მოიყენება არითმეტიკულ გამოსახულებაში, მაშინ ის ავტომატურად იქნება მიყვანილი რიცხვით ტიპზე. თუ კი ასეთი სტრიქონული, ტექსტური ცვლადი შეიცავს რიცხვით მნიშ­ვნელობასთან ერთად სხვა სიმბოლოებს, მაშინ მისი მნიშვნელობის დასაწყისიდან გამო­იყოფა მაქსიმალურად შესაძლებელი რიცხვი.

ანალოგიურად რიცხვითი ცვლადები ავტომატურად გარდაიქმნებიან სტრიქონულში, თუ კი ოპერაციაში არის აუცილებელი სტრიქონული ცვლადის არსებობა - ვთქვათ საჭიროა ბრაუზერის ფანჯარაში გამოტანა. თუ კი მოლოდინია ლოგიკურ ტიპზე, პირობითი ოპერატორის კვალობაზე, მაშინ რიცხვები, რომელიც ნულის ტოლია; ცარიელი სტრიქონი; სტრიქონი რომელიც შეიცავს ”0” (ნულს); ცარიელი მასივები და ობიექტები; ასევე NULL ავტომატურად მიიყვანება FALSE მნიშვნელობამდე, სხვა დანარჩენ შემთხვევაში კი გვაქვს მნიშვნელობა -TRUE.

დესკრიპტორები ყოველთვის განიხილება როგორც TRUE, ამიტომაც დესკრიპტორის კორექ­ტული გამოყოფისას is\_resource() ფუნქციის გამოყენების ნაცვლად, როგორც ეს მოხდა ლის­ტინგ 7-ში, პირობით ოპერატორში პირდაპირ დესკრიპტორი ჩაისმება. ლოგიკური ტიპის გარდაქმნა სტრიქონულ ცვლადში ხდება შემდეგნაირად: TRUE გარდაიქმნება ”1”-ში, ხოლო FALSE ცარიელ სუტიქონში - ””. ლოგიკური ტიპის გარდაქმნა რიცხვითში ხდება შემდეგნა­ირად: TRUE – 1-ში, FALSE – 0-ში.

TRUE ყოველთვის გამოიტანება როგორც 1-თი, FALSE კი როგორც ცარიელი სტრიქონი.

მოვიყვანოთ ტიპების არაცხადად მიყვანის დემონსტრირება

ლისტინგში 8-ში - ტიპების არაცხადი დაყვანა

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

 $str = "12.6";

 $number = 3 + $str;

 echo $number; // 15.6

 echo "12wet56.7" + 10; // 12 + 10 = 22

 echo "<br>";

 echo 14 + "თოთხმეტი"; // 14 + 0 = 14

 echo "<br>";

 // ვაცხადებთ ნულის ტოლ ნამდვილ რიცხვს

 $float = 0.0;

 if($float) // FALSE

 echo 'ცვლადი $float განიხილება როგორც TRUE';

 $str1 = "მოგესალმებით ყმაწვილებო!";

 if($str1) // TRUE

 echo 'ცვლადი $str1 განიხილება როგორც TRUE<br>';

 // ვღებულობთ მბმს MySQL -თან დაკავშირების დესკრიპტორს

 $dbcnx = mysql\_connect("localhost", "root", "");

 // ვამოწმებთ დესკრიპტორის კორექტულ გამოყოფას

 if($dbcnx)

 {

 // დესკრიპტორი წარმატებითაა მიღებული და ვახორციელებთ

 // შემდეგ მოქმედებას

 echo get\_resource\_type($dbcnx); // mysql link

 }

 echo TRUE; // "1"

 echo FALSE // ""

?>

</BODY>

</HTML>

ლისტინგ 8-ის შესრულებით ვღებულობთ

15.622
14
ცვლადი $str1 განიხილება როგორც TRUE
mysql link1

**ცხადი სახით ტიპებზე მიყვანა**

ცვლადების ტიპების არაცხადად დაყვანის გარდა არსებობს ცვლადების ტიპების ცხადი სახით მიყვანის რამდენიმე ვარიანტი.

ერთ-ერთის შემთხვევაში, გამოიყენება მრგვალი ფრჩხილები, რომელშიც ჩაიწერება იმ ტიპის

 დასახელება, რომელამდეც მიგვყავს ფრჩხილების შემდეგ ჩაწერილი ცვლადი. შევნიშნოთ,

 რომ როცა ნამდვილი რიცხვი მიგვყავს მთელ ტიპამდე მისი წილადი ნაწილის დამრგვალება

არ ხდება, წილადი ნაწილი უბრალოდ იგნორირდება. ნამდვილი რიცხვის მთელ რიცხვამდე

დამრგვალებისთვის,დამრგვალების მათემატიკური წესების გათვალისწინებით, უნდა გა­-

მოვიყენოთ ფუნქცია round().

განვიხილოთ მაგალითი ნამდვილი რიცხვის მთელ ტიპამდე მიყვანისა.

ლისტინგი 9 ნამდვილი რიცხვის მთელ ტიპამდე მიყვანა

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

 <?php

// ვაცხადებთ ნამდვილ რიცხვს, არაცხადი სახით

$float = 4.863;

// მივიყვანთ ცვლად $float -ს

// მთრლ ტიპამდე

$number = (int)$float;

echo $number; // 4

echo '<br>';

// ვამრგვალებთ წესების დაცვით

$number1= round($float);

echo $number1; // 5

?>

</BODY>

</HTML>

ლისტინგი 9-ს შესრულების შემდეგ მივიღებთ

4

5

ქვემოთ მოყვანილ ცხრილში მოცემულია მრგვალი ფრჩხილების გამოყენების ყველა შესაძლო ვარიანტი:

|  |  |
| --- | --- |
| გარდაქმნა | აღწერა |
| $var = (int) $var;  | მთელ int ტიპზე მიყვანა |
| $var = (integer) $var;  | მთელ int ტიპზე მიყვანა |
| $var = (bool) $var;  | ლოგიკურ boolean ტიპზე მიყვანა |
| $var = (boolean) $var;  | ლოგიკურ boolean ტიპზე მიყვანა |
| $var = (float) $var;  | ნამდვილ double ტიპზე მიყვანა |
| $var = (double) $var;  | ნამდვილ double ტიპზე მიყვანა |
| $var = (real) $var;  | ნამდვილ double ტიპზე მიყვანა |
| $var = (string) $var;  | სტრინგულ, ტექსტურ string ტიპზე მიყვანა |
| $var = (binary) $var;  | ბინარულ სტრიქონამდე მიყვანა |
| $var = (array) $var; | მასივამდე მიყვანა |
| $var = (object) $var; | ობიექტამდე მიყვანა |

ტიპების ცხადი გარდაქმნის გამოყენების მაგალითად მოვიყვანოთ რიცხვის ლუწობის დასადგენად: შესამოწმებელი რიცხვი ორჯერ იყოფა ორზე, ამასთამ პირველ შემთხვევაში ის გარდაიქმნება მთელ რიცხვად, მეორე შემთხვევაში კი ნამდვილ რიცხვად. მიღებული შედეგების სხვაობით დგინდება - თუ სხვაობა ნულის ტოლია, if ოპერატორში ის იძლევა FALSE -ს - შესამოწმებელი რიცხვი არის ლუწი, ხოლო სხვა შემთხვევაში - როცა სხვაობა 0,5, if ოპერატორში ის იძლევა TRUE -ია, მაშინ კენტია.

ლისტინგი 10 რიცხვის ლუწობა-კენტობის დადგენა ცვლადების ტიპების ცხადი მითითებით

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

<?php

<?php

// გამოვაცხადოთ რიცხვი

$number = 15;

$flag = (float)($number/2) - (int)($number/2);

// განვსაზღვროთ რიცხვის სტატუსი

if($flag) // TRUE, რადგანაც $flag == 0.5

echo "რიცხვი $number კენტია";

else

echo "რიცხვი $number ლუწია";

$str1= (binary) "1234.5";

$str 2= b"1234.5";

echo "$str1<br>$str 2""

?>

</BODY>

</HTML>

ლისტინგი 10-ის გაშვებით მივიღებთ:

**რიცხვი 15 კენტია
1234.5
1234.5**

მრგვალი ფრჩხილებით ცვლადების ტიპების გარდაქმნის გარდა არის კიდევ რიგი სპეციალური ფუნქციები, რომლებიც იგივე მიზნებისთვის გამოიყენება, მაგალითად

ცხრილი 2.

|  |  |
| --- | --- |
| ფუნქცია | აღწერა |
| settype($*var*,$*type*) | გარდაქმნის $*var*  ცვლადს იმ ტიპში რომელიც მითითებულია $*type* პარამეტრში - მან შეიძლება მიიღოს ერთ-ერთი შემდეგი მნიშვნელობა:"boolean", "bool", "integer", "int", "float", "string", "array", "object" და "null". ფუნქცია აბრუნებს TRUE, თუ გარდაქმნა წარმატებულია და FALSE წინააღმდეგ შემთხვევაში |
| floatval($*var*)  | გარდაქმნის $*var* ცვლადს ნამდვილ ტიპზე float |
| doubleval($*var*)  | ფუნქცია floatval() -ის სინონიმო |
| intval($*var*[, $*base*]) | გარდაქმნის $*var*  ცვლადს მთელ ტიპამდე intფუძით $*base* |
| strval($*var*)  | გარდაქმნის $*var*  ცვლადს სტრიქონულ ტიპამდე  |

ვანახოთ ამ ფუნქციების გამოყენების მაგალითი

ლისტინგ 11 ტიპების მიყვანა settype

<!DOCTYPE HTML>

<HTML>

<HEAD></HEAD>

<BODY>

<?php

 // გამოვაცხადოთ სტრიქონული ცვლადი

 $str = "5wet";

 // მივიყვანოთ სტრიქონი მთელ ტიპამდე

 settype($str, "integer");

 echo $str; // 5

 // სტრიქონის გადაყვანა ბრაუზერში

 echo "<br>";

 // ლოგიკური ცვლადის გამოცხადება

 $flag = TRUE;

 // ლოგიკური ცვლადის მიყვანა სტრიქონამდე

 settype($flag, "string");

 echo $flag; // 1

 echo "<br>";

 $a=56; //მთელი ტიპის ცვლადის გამოცხადება

 echo gettype($a).'<br>';//ვადგენტ თუ რა ტიპისაა $a

 settype ($a,'float');// გარდავქნით $a-ს ტიპს ნამდვილ ტიპში

 echo gettype($a).'<br>';// ეხლა ვადგენთ გარდაქმნილი $a-ს ტიპს

 echo "<br>"; echo intval(42); // 42

 echo "<br>"; echo intval(4.2); // 4

 echo "<br>"; echo intval('42'); // 42

 echo "<br>"; echo intval('+42'); // 42

 echo "<br>"; echo intval('-42'); // -42

 echo "<br>"; echo intval(042); // 34

 echo "<br>"; echo intval('042'); // 42

 echo "<br>"; echo intval(1e10); // 1410065408

 echo "<br>"; echo intval('1e10'); // 1

 echo "<br>"; echo intval(0x1A); // 26

 echo "<br>"; echo intval(42000000); // 42000000

 echo "<br>"; echo intval(420000000000000000000); // 0 მიენიჭა გადავსების გამო მთელ ტიპს

 echo "<br>"; echo intval(-420000000000000000000); // 0 მიენიჭა გადავსების გამო მთელ ტიპს

 echo "<br>"; echo intval('420000000000000000000'); // 2147483647 - ეს მთელი რიცხვის მაქსიმალური მნიშვნელობაა, რომელიც მიენიჭა გადავსების გამო სტრიქონულ ცვლადში

 echo "<br>"; echo intval('-420000000000000000000'); // -2147483648 - ეს მთელი რიცხვის მინიმალური მნიშვნელობაა, რომელიც მიენიჭა გადავსების გამოს ტრიქონულ ცვლადში

 echo "<br>"; echo intval(42, 8); // 42 თანრიგის მითითება საჭიროა სტრიქონული ცვლადისთვის

 echo "<br>"; echo intval('42', 8); // ’42’ წარმოდგენილია 8-ით სისტემაში, ვღებულობთ მის 10-ით ექვივალენტს - 34

 echo "<br>"; echo intval('42', 9); // ’42’ წარმოდგენილია 9-ით სისტემაში, ვღებულობთ მის 10-ით ექვივალენტს - 38

 echo "<br>"; echo intval('42', 11); // ’42’ წარმოდგენილია 11-ით სისტემაში, ვღებულობთ მის 10-ით ექვივალენტს - 46

 echo "<br>"; echo intval('42', 16); // ’42’ წარმოდგენილია 16-ით სისტემაში, ვღებულობთ მის 10-ით ექვივალენტს - 66

 echo "<br>"; echo intval(0x42); // ’42’ წარმოდგენილია 16-ით სისტემაში, ვღებულობთ მის 10-ით ექვივალენტს - 66

?>

</BODY>

</HTML>

ლისტინგ 11-ის გაშვების შემდეგ გვაქვს, localhost-ზეა:

**5 1 integer double 42 4 42 42 -42 34 42 1410065408 1 26 42000000 0 0 2147483647 -2147483647 42 34 38 46 66 66**

**სერვერზე -**

5 1 integer double 42 4 42 42 -42 34 42 10000000000 1 26 42000000 0
-9223372036854775808 9223372036854775807 -9223372036854775808
42 34 38 46 66 66

ლისტინგ 11-ის კომენტარებში მოცემულია intval() ფუნქციის მუშაობის ნიუანსები