**ლექცია № 5 PHP ბეჭდვა. ტიპები**

PHP-ში ინფორმაციის გამოტანის ორი ძირითადი საშუალება არსებობს - **echo** და **print**.

echo და print ოპერატორები PHP-ში არის რამდენიმე განსხვავება:

* **echo**-ს შეუძლია გამოიტანოს ერთი ან რამდენიმე სტრიქონი ინფორმაციის;
* **print**-ს შეუძლია გამოიტანოს მხოლოდ ერთი სტრიქონი ინფორმაციის და აბრუნებს მნიშვნელობას 1-ს, თუ ეს ოპერაცია წარმატებით შესრულდა და 0-ს, თუ კი არა;

აქედან გამომდინარე, რადგანაც print-ი აბრუნებს მნიშვნელობას იგი უფრო ნელია echo-სთან შედარებით.

*შენიშვნა* - „სტრიქონი“ ამ შემთხვევაში უნდა გაგებულ იქნეს, როგორც ენის ანბანის სიმბოლოების მიმდევრობად - ისინი შეიძლება იყვნენ კონსტანტები ან ცვლადები.

echo -ს სინაქსისი ასე გამოისახება:

echo(str1[,str2…]) სადაც str1,str2 სტრიქონებია, print -ს სინაქსისი კი ასე გამოისახება

print(str1).

**echo არის** PHPენის კონსტრუქცია და მისი გამოყენება შეიძლება, როგორც ფრჩხილებით, ასევე მათ გარეშე: echo ან echo(). იგივე ითქმის print ოპერატორზე - print ან print(). შევნიშნოთ, რომ () არსებობა დამახასიათებელია ფუნქციებისთვის.

ლისტინგი 1 echo-ს სინტაქსის რეალიზება

<!DOCTYPE html>

<html>

<body>

<?php

$str2="ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.";// სტრიქონული ცვლადი

$str3="ვარდები არ არიან, მაგრამ რა მევარდება?";

$str4="სულო, რა გემართება? გულო, რა მოგივიდა?";

$str5=1927;

$str6="გალაკტიონ ტაბიძე";

echo "შემოდგომის ყვავილებს დიდებასთან მივიტან!",$str2, $str3, $str4, $str5, $str6;

?>

</body>

</html>

ლისტინგ 1-ის შესრულებით http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_1\_sintaqsis/ მივიღებთ:

**შემოდგომის ყვავილებს დიდებასთან მივიტან!ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.ვარდები არ არიან, მაგრამ რა მევარდება?სულო, რა გემართება? გულო, რა მოგივიდა?1927გალაკტიონ ტაბიძე**

აღსაქმელად ეს ჩანაწერი არც თუ ისე სასიამოვნოა. გამოსატანი ინფორმაციის სტრუქტურირებისთვის შესაძლებელია გამოყენებულ იქნენ html-ის ტეგები, რომელიც ბრჭყალებში უნდა მოვათავსოთ და მძიმეებით გამოვყოთ. მაგალითად

ლისტინგი 2 - echo-ში html-ის ტეგების ჩასმა

<!DOCTYPE html>

<html>

<body>

<?php

$str2="ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.";// სტრიქონული ცვლადი

$str3="ვარდები არ არიან, მაგრამ რა მევარდება?";

$str4="სულო, რა გემართება? გულო, რა მოგივიდა?";

$str5=1927;

$str6="გალაკტიონ ტაბიძე";

echo "შემოდგომის ყვავილებს დიდებასთან მივიტან!<br>",$str2, '<br>',$str3,'<br>', $str4,'<br>',$str5,'<br>',$str6;

?>

</body>

</html>

ლისტინგ 2-ის შესრულებით http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_2\_sintaqsis\_html/ მივიღებთ:

**შემოდგომის ყვავილებს დიდებასთან მივიტან!  
ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.  
ვარდები არ არიან, მაგრამ რა მევარდება?  
სულო, რა გემართება? გულო, რა მოგივიდა?  
1927  
გალაკტიონ ტაბიძე**

შესაძლებელია ლისტინგ 2-ში ფერებიც შემოვიტანოთ და სხვა ტეგები, მაგალითად ლისტინგი 3, რომელიშიც ყველა სტრიქონი იგივეა მეთერთმეტე სტრიქონის გარდა, რომელიც ასე გამოისახება -

**echo "შემოდგომის ყვავილებს დიდებასთან მივიტან!",'<br>',$str2, '<br>',$str3,'<br>', $str4,'<br>','<br>',$str5,'<br><font color="red"><b>',$str6,'</b></font>';**

ლისტინგ 3-ის შესრულებით http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_3\_sintaqsis\_html\_color/ მივიღებთ:

შემოდგომის ყვავილებს დიდებასთან მივიტან!  
ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.  
ვარდები არ არიან, მაგრამ რა მევარდება?  
სულო, რა გემართება? გულო, რა მოგივიდა?  
  
1927  
**გალაკტიონ ტაბიძე**

ავღნიშნოთ, რომ echo-ში ჩვენ სტრიქონებს შორის მოვათავსეთ მძიმე, იგივე შედეგს მივიღებთ, თუ კი მძიმის ნაცვლად გამოვიყენებთ წერტილს ”**.** ” - ოპერატორ წერტილს, რომლის ს აშუალებითაც ცვლადების გაერთიანება - *კონკატენაცია* შეიძლება. ამის მაგალითად ლისტინგ 4-ში ისევ მეთერთმეტე სტრიქონი შევცვალოთ შემდეგით -

echo "შემოდგომის ყვავილებს დიდებასთან მივიტან!".'<br>'.$str2. '<br>'.$str3.'<br>'. $str4.'<br>'.'<br>'.$str5.'<br><font color="red"><b>'.$str6.'</b></font>';

ლისტინგ 4-ის გაშვებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_4\_echo\_konkatenacia/ იმავე შედეგს მივიღებთ, რაც ლისტინგ 2-ის, ლისტინგ 3-ის გაშვების შედეგად.

განვიხილოთ ეხლა რამდენიმე ტრიქონიანი უნფორმაციების გამოტანის ჰერედოკ (heredoc) სინტაქსისი - <<< . heredoc-სინტაქსისი Perl იდანაა აღებული.

განვიხილოთ მაგალითი

ლისტინგი 5, heredoc-სინტაქსისი

<!DOCTYPE html>

<html>

<body>

<?php

echo <<< Galaqtioni

შემოდგომის ყვავილებს დიდებასთან მივიტან!

ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.

ვარდები არ არიან, მაგრამ რა მევარდება?

სულო, რა გემართება? გულო, რა მოგივიდა?

1927

გალაკტიონ ტაბიძე

Galaqtioni

?>

</body>

</html>

ლისტინგ 5-ის გაშვებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_5\_heredoc/ იმავე შედეგს მივიღებთ, რაც ლისტინგ 1-ის გაშვების შედეგად -შედეგი გამოვიდა არასტრუქტურიზებული.

Galaqtioni არის ჭდე ამ მრავალსტრიქონიანი ინფორმაცია, მისით უნდა მთავრდებოდეს ეს მრავალსტრიქონიანი ინფორმაცია, თანაც მხოლოს ცალკე სტრიქონად. ჭდე არ უნდა

heredoc-სინტაქსისში შესაძლებელის ცვლადების გამოყენება და მაშინ იბეჭდება მათი მნიშვნელობები. ანუ heredoc-სინტაქსისი ამ მხრივ ორმაგი ბრჭყალების პრინციპით მუშაობს

ლისტინგი 6, heredoc-სინტაქსისი ცვლადებით

<!DOCTYPE html>

<html>

<body>

<?php

$str2="ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.";// სტრიქონული ცვლადი

$str3="ვარდები არ არიან, მაგრამ რა მევარდება?";

$str4="სულო, რა გემართება? გულო, რა მოგივიდა?";

$str5=1927;

$str6="გალაკტიონ ტაბიძე";

echo <<< Galaqtioni

შემოდგომის ყვავილებს დიდებასთან მივიტან!

$str2

$str3

$str4

$str5

$str6

Galaqtioni

?>

</body>

</html>

ლისტინგ 6-ის გაშვებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_5\_heredoc\_1/ იმავე შედეგს მივიღებთ, რაც ლისტინგ 1-ის და ლისტინგ 5-ის გაშვების შედეგად -შედეგი გამოვიდა არასტრუქტურიზებული.

შევნიშნოთ, რომ heredoc-სინტაქსისში არანაირი ბრჭყალები არ გვჭირდება, არც კონკატენა­ცია წერტილით და არც მძიმის გამოყენება. მონაცემთა სტრუქტურირებული გამოყვანისას html -ის ტეგები გამოიყენება ბრჭყალების გარეშე, მაგალითად

ლისტინგი 7

<!DOCTYPE html>

<html>

<body>

<?php

echo <<< Galaqtioni

შემოდგომის ყვავილებს დიდებასთან მივიტან! <br>

ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.<br>

ვარდები არ არიან, მაგრამ რა მევარდება?<br>

სულო, რა გემართება? გულო, რა მოგივიდა?<br>

1927<br><br>

<b><i>გალაკტიონ ტაბიძე</b></i>

Galaqtioni

?>

</body>

</html>

ლისტინგ 7-ის გაშვებისას მივიღებთ http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_5\_heredoc\_2/ შედეგს:

შემოდგომის ყვავილებს დიდებასთან მივიტან!   
ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.  
ვარდები არ არიან, მაგრამ რა მევარდება?  
სულო, რა გემართება? გულო, რა მოგივიდა?  
1927  
  
***გალაკტიონ ტაბიძე***

heredoc-სინტაქსისი გამოიყენება ტექსტური ცვლადის შესაქმნელად, ე.ი. ჩვენ შეგვიძლია ლისტინგ 7-ში შეგვექმნა ტექსტური ცვლადი, ვთქვათ $var\_heredoc და შემდეგ გამოგვეტანა ის ბეჭდვაზე, რაც მოცემულია ქვემოთ მოყვანის მაგალითში:

ლისტინგი 8 ცვლადი heredoc-სინტაქსისით

<!DOCTYPE html>

<html>

<body>

<?php

$var\_heredoc = <<< Galaqtioni

შემოდგომის ყვავილებს დიდებასთან მივიტან! <br>

ერთი შორი ოცნება მახსოვს ძველ მოტივიდან.<br>

ვარდები არ არიან, მაგრამ რა მევარდება?<br>

სულო, რა გემართება? გულო, რა მოგივიდა?<br>

1927<br><br>

<b><i>გალაკტიონ ტაბიძე</b></i>

Galaqtioni;

echo $var\_heredoc;

?>

</body>

</html>

ლისტინგ 8-ის გაშვებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_5\_heredoc\_3/ მივიღებთ იმავე შედეგს რაც ლისტინგ 7-ის დროს:

ჩვენ შეგვიძლია echo-ში გამოვიყენოთ ფიგურული ფჩხილები, რომლის საშუალებით შესაძლებელის ცვლადის გამონცალკავება ჰარის გარეშეც. მაგალითად

ლისტინგი 9 echo ფიგურული ფრჩხილით

<!DOCTYPE html>

<html>

<body>

<?php

$kala ="კალა";

echo "{$kala}თბურთი<br>{$kala}ძე<br>{$kala}პოტი";

?>

</body>

</html>

echo ფიგურული ფრჩხილით განსაკუთრებით ეფექტურია, როცა ტექსტური სტრიქონი ლათინური ანბანითაა აგებული და ცვლადის უშუალო გაგრძებაა ეს ლათინური ლიტერები. მაგალითად ლისტინგ 9-ის მეშვიდე სტრიქონი ჩაიწერებოდა ასე:

**echo "{$kala}TburTi<br>{$kala}Ze<br>{$kala}poti";**

რაც გაცილებით უკეთესია, ამ ჩანაწერისა:

**echo "$kalaTburTi<br>$kalaZe<br>$kalapoti";**

ამ უკანასკნელ შემთხვევაში PHP ინტერპრეტარი ვერ გაარკვევდა ცვლადებს $kalaTburTi, $kalaZe, $kalapoti და ვერაფერს ვერ გამოიტანდა

ლისტინგ 9-ის გაშვებისას მივიღებთ <http://www.kereseli.besaba.com/PHP/laboratoriuli_5/lab_5_6_echo_fig_fr/> შედეგს:

**კალათბურთი  
კალაძე  
კალაპოტი**

იმ შემთხვევაში, როცა PHP ბლოკში მხოლოდ ერთი - გამოტანის echo ოპერატორია, მაშინ PHP-ის დესკრიპტორები მოკლე სახით ჩაიწერება, მაგალითად:

ლისტინგი 10 - PHP-ის მოკლე დესკრიპტორი

<!DOCTYPE html>

<html>

<body>

<?="ეს დესკრიპტორის მოკლე ფორმაა" ?>

</body>

</html>

ლისტინგ 10-ის გაშვების შემდეგ http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_5\_mokle\_diskript/ მივიღებთ:

**ეს დისკრიპტორის მოკლე ფორმაა**

თუ კი გაინტერესებთ PHP-ს. Apache-ს, MySQL-ს და სხვათა სპეციფიკაციების გასაცნობად გამოსაყენებელია ფუნქცია phpinfo()

ლისტინგი 11 ინფორმაცია PHP-ზე

<!DOCTYPE html>

<html>

<body>

<?php

phpinfo();

?>

</body>

</html>

ლისტინგ 11-ის გაშვების შემდეგ http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_7\_info\_php/ მივიღებთ საჭირო ინფორმაციას PHP-ზე და სხვა

შეიძლება თუ არა გამოტანის ოპერატორებში ცვლადებზე, სტრინგებზე, კონსტანტებზე მოქმედებების ჩაწერა და მათი შესრულება? პასუხი დადებითია, მაგრამ სანამ ამ კითხვას დაწვრილებით გავცემთ პასუხს უფრო დეტალურად განვიხილოთ ცვლადები და მისი ტიპები.

**მონაცემთა ტიპები**

როგორც უკვე ვისაუბრეთ PHP ენა არის სუსტადტიპიზირებული, ანუ ამ ენის ცვლადები არ საჭიროებენ ტიპების მკაცრ მითითებას მათი გამოცხადებისას, ხოლო პროგრამის- სკრიპტის შესრულების მიმდინარეობისას, ცვლადის ტიპი შეიძლება შეცვლილ იყოს არაცხადი სახით - სპეციალურ გარდაქმნას არ საჭიროებენ. მაგალითად თუ ცვლადი თავიდან იყო გამოცხადებული როგორც სტრინგი - ტექსტური, შემდგომში შეიძლება გამოყენებულ იყოს არითმეტიკულ ოპერაციებში, წარმოდგენილი როგორც ლოგიკური ცვლადი, და ბოლოს და ბოლოს მას შეიძლება მიენიჭოს როგორც მნიშვნელობა - ობიექტი. ეს თვისებები აძლევს საშუალებას დამპროგრამირებელს არც ისე დიდი ყურადღება დაუთმოს მონაცემების ტიპებს, მაგრამ, მაინც ცვლადები ტიპებზე გარკვეული შეზღუდვები არსებობენ და მასში კარგად უნდა ვერკვეოდეთ.

PHP-ში არის ცვლადების შემდეგი ტიპები: ლოგიკური, მთელი, ნამდვილი, ტექსტური, მასსივი, ობიექტი, რესურსული, სპეციალური. მოვიყვანოთ ტიპები ცხრილის სახით

|  |  |
| --- | --- |
| **მონაცემთა ტიპები** | **აღწერა** |
| boolean | ლოგიკური ტიპი, რომელსაც შეუძლია მიიღოს მხოლოდ ორი მნიშვნელობა:TRUE (ჭეშმარიტი) და FALSE (მცდარი). |
| integer | მთელი რიცხვი, რომმლის თანრიგთა რაოდენობა კომპიუტერის არქიტექტურაზეა დამოკიდებული.  თუ ოპერაციული სისტემა თანრიგობა 32 ბიტია, მაშინ მთელ რიცხვებს შეუძლიათ მიიღო მნიშცნელობები შემდეგი დიაპაზონიდან: **-2 147 483 648**-დან **2 147 483 647**-მდე.  თუ ოპერაციული სისტემა თანრიგობა 64 ბიტია, მაშინ მთელ რიცხვებს შეუძლიათ მიიღო მნიშცნელობები შემდეგი დიაპაზონიდან: **-9 223 372 036 854 775 808**-დან **-9 223 372 036 854 775 807**-მდე. |
| int64 | მთელი რიცხვი, რომლის თანრიგთა რაოდენობა დამოკიდებული არაა კომპიუტერის არქიტექტურაზე და მუდამ 64 ბიტის ტოლია.  მთელ რიცხვებს შეუძლიათ მიიღო მნიშცნელობები შემდეგი დიაპაზონიდან: -9 223 372 036 854 775 808-დან -9 223 372 036 854 775 807-მდე. |
| double (ან float) | ნამდვილი რიცხვი, რომლის აბსოლუტური მნიშვნელობა მოთავსებულია დიაპაზონში: **1,7x 10-308** -დან **1,7x 10+308** -მდე |
| string | სტრიქონული, ტექსტური ტიპი. მაშ შეუძლია შეინახის ნებისმიერი, კომპიუტერისთვის დასაშვები, მოცულობის ინფორმაცია |
| array | მასივი - რამდენიმე ტიპის ცვლადების გაერთიანება ერთი სახელის - იდენტიფიკატორის ქვეშ. მასივის ცალკეული ელემენტებისადმი მამართვა შესაძლებელია მასივის ინდექსის საშუალებით |
| object | ობიექტი. გვაძლევს საშუალებას გავაერთიანოთ რამდენიმე სხვადასხვა ტიპის ცვლადები და მათი დამუშავების მეთოდები |
| resource | დესპრიპტორი, რომელიც იძლევა საშუალებას მოახდინოს ოპერირება ამა თუ იმ რესურსზე, რომელთა წვდომა შესაძლებელია ფუნქციათა ბიბლიოთეკის საშუალებით. დესპრიპტორი გამოიყენება ფაილებთან, მონაცემთა ბაზებზე, დინამიურ გამოსახულებებთან და სხვებზე მუშაობისას |
| NULL | სპეციალური ტიპი, რომელიც მიანიშნებს იმას, რომ ცვლადი არ იყო ინიციალიზირებული |

**PHP Booleans**

Booleans შეიძლება ჭეშმარიტი ან მცდარი.

$x=true;  
$y=false;

Booleans ხშირად გამოიყენება პირობის ტესტირებისას.

**მთელი ტიპი**

მთელი არის რიცხვი მძიმის გარეშე - წილადი ნაწილის გარეშე.

მთელი რიცხვების წესები:

* მთელი უნდა შედგებოდეს არანაკლებ ერთი ციფრისგან (0-9) მაინც
* მთელი არ შეიცავს მძიმეს ან შორის (ჰარის)
* მთელს არ უნდა ჰქონდეს ათობითი წერტილი
* მთელი შეიძლება იყოს როგორც დადებითი ასევე უარყოფითი
* მთელი შეიძლება მოცემულ იქნეს სამი ფორმით: ათობითი, რვაობითი და თექვსმეტობითი

მთელი ტიპის ცვლადები ათობით სისტემაში მოიცემა ჩუმათობის პრინციპით, მაგრამ PHP-ში შეიძლება მთელი რიცხვები მოცემულ იქნენ რვაობით და თექვსმეტობით სისტემაში. ვთქვათ $int1 -ს მივანიჭეთ მთელი რიცხვი 123, რომელიც ასე გამოისახება: $int1=123; რადგან ეს რიცხვი დადებითია მის ნიშანს ”+” არ ვწერთ, ჩუმათობით იგულისხმება.

იმისთვის, რომ $int2 -ს მივანიჭოთ -123, ჩავწერთ $int2=-123; რადგან ეს რიცხვი უარყოფითია მის ნიშანს ”-” აუცილებლად ვწერთ.

თუ კი რიცხვი 123 არის რვაობით სისტემაში, მაშინ იმისთვის რომ იგი მივანიჭოთ $int3 ცვლადს, მას წინ უნდა დავუწეროთ ნული: $int3=0123; (123)8=1x82+2x81+3x80=64+16+3=(83)10

თექვსმეტობითი სისტემის მთელი რიცხვის 2AF-ის მინიჭებისთვის $int4 -ისათვის საჭიროა გამოვიყენოთ პრეფიქსი 0x, მაგალითად $int4=0x2AF; 10 – A: 11 – B; 12-C; 13 – D; 14 – E; 15 –F;

(2AF)16=2 x162+ A x161+ F x160=2 x 256 + 10 x 16 + 15 x1= 512+160+15=(687)10

ლისტინგი 12 - მთელი ტიპი

<!DOCTYPE html>

<html>

<body>

<?php

$int=+123;// ათობითი დადებითი რიცხვი

echo "$int ათობითი დადებითი რიცხვი<br>";

$int1=123; // ათობითი დადებითი რიცხვი

echo "$int1 ათობითი დადებითი რიცხვი<br>";

$int2=-123;// ათობითი რიცხვი

echo "$int2 ათობითი უარყოფითი რიცხვი<br>";

$int3=0123;//რვაობითი 123, რომელიც ათობითი 83-ის ექვივალენტურია

echo "$int3 რვაობითი 123, რომელიც ათობითი 83-ის ექვივალენტურია<br>";

$int4=0x2AF;//თექვსმეტობითი 2AF, რომელიც ათობითი 687-ის ექვივალენტურია

echo "$int4 თექვსმეტობითი 2AF, რომელიც ათობითი 687-ის ექვივალენტურია<br>";

?>

</body>

</html>

ლისტინგ 12-ის გაშვების შემდეგ http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_8\_mTeli\_tipi/ მივიღებთ:

**123 ათობითი დადებითი რიცხვი  
123 ათობითი დადებითი რიცხვი  
-123 ათობითი უარყოფითი რიცხვი  
83 რვაობითი 123, რომელიც ათობითი 83-ის ექვივალენტურია  
687 თექვსმეტობითი 2AF, რომელიც ათობითი 687-ის ექვივალენტურია**

თუ კი მთელი რიცხვის გადავსება შესაძლებელია დაპროგრამების სხვა ენებში, PHP-ში integer გადავა ან int64 -ში ან double (float) -ში.

PHP-ში არის ფუნქცია **var\_dump()**, რომელიც გვაძლევს საშუალებას დაგვიბრუნოს (დაგვიბეჭდოს) ცვლადის ტიპი და მისი მნიშვნელობა. var\_dump()-ის ფრჩხილებში იწერება შესამოწმებელი ცვლადის სახელი(ები)-იდენტიფიკატორ(ებ)ი ან მნიშვნელობა/მნიშვნელობები

ლისტინგი 13 - მთელი ტიპი var\_dump()-ით

<!DOCTYPE html>

<html>

<body>

<?php

$int=+123;// ათობითი დადებითი რიცხვი

var\_dump($int);

echo "$int ათობითი დადებითი რიცხვი<br>";

$int1=123; // ათობითი დადებითი რიცხვი

var\_dump($int1);

echo "$int1 ათობითი დადებითი რიცხვი<br>";

$int2=-123;// ათობითი რიცხვი

var\_dump($int2);

echo "$int2 ათობითი უარყოფითი რიცხვი<br>";

$int3=0123;//რვაობითი 123, რომელიც ათობითი 83-ის ექვივალენტურია

var\_dump($int3);

echo "$int3 რვაობითი 123, რომელიც ათობითი 83-ის ექვივალენტურია<br>";

$int4=0x2AF;//თექვსმეტობითი 2AF, რომელიც ათობითი 687-ის ექვივალენტურია

var\_dump($int4);

echo "$int4 თექვსმეტობითი 2AF, რომელიც ათობითი 687-ის ექვივალენტურია<br>";

var\_dump($int,$int1,$int2,$int3,$int4);

?>

</body>

</html>

ლისტინგ 13-ის გაშვების შემდეგ http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_9\_mTeli\_tipi\_dump/ მივიღებთ

**int(123) 123 ათობითი დადებითი რიცხვი**  
**int(123) 123 ათობითი დადებითი რიცხვი  
int(-123) -123 ათობითი უარყოფითი რიცხვი  
int(83) 83 რვაობითი 123, რომელიც ათობითი 83-ის ექვივალენტურია  
int(687) 687 თექვსმეტობითი 2AF, რომელიც ათობითი 687-ის ექვივალენტურია  
int(123) int(123) int(-123) int(83) int(687)**

**ნამდვილი ტიპი**

ნამდვილი ანუ მცოცავი მძიმით, არის რიცხვი რომელსაც აქვს ათობითი წერტილი ან წარმოდგენილია ექსპონენციალური ფორმით. ზოგჯერ ნამდვილი რიცხვის მოცემის ფორმებს უწოდებენ: სტანდარტულს და მეცნიერულს.

მეცნიერული ფორმით ნამდვილი რიცხვის ჩაწერისას გვაქვს მანტისა (მთელი და წილადი ნაწილი), და რიგი - რომელიც იწყება ასო E (მაღალ ან დაბალ რეგისტრში) და მთელი დადებითი ან უარყოფითი რიცხვისგან. მაგალითად 2.5 , 4.3 E12.

ლისტინგ 14 - ნამდვილი რიცხვი

<!DOCTYPE html>

<html>

<body>

<?php

$x = 19.1365;//სტანდარტული ფორმით ჩაწერილი ნამდვილი რიცხვი

echo '19.1365 სტანდარტული ფორმით ჩაწერილი ნამდვილი რიცხვი ';

var\_dump($x);

echo "<br>";

$x = 22.14e3;//მეცნიერული ფორმით ჩაწერილი ნამდვილი რიცხვი

echo '22.14e3 მეცნიერული ფორმით ჩაწერილი ნამდვილი რიცხვი ';

var\_dump($x);

echo "<br>";

$x = 12E-5;//მეცნიერული ფორმით ჩაწერილი ნამდვილი რიცხვი

echo '12E-5 მეცნიერული ფორმით ჩაწერილი ნამდვილი რიცხვი ';

var\_dump($x);

?>

</body>

</html>

ლისტინგ 14-ის გაშვებით http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_10\_namdvili/ მივიღებთ

**19.1365 სტანდარტული ფორმით ჩაწერილი ნამდვილი რიცხვი float(19.1365)   
22.14e3 მეცნიერული ფორმით ჩაწერილი ნამდვილი რიცხვი float(22140)   
12E-5 მეცნიერული ფორმით ჩაწერილი ნამდვილი რიცხვი float(0.00012)**

**სტრიქონული (ტექსტური) ტიპი**

სტრიქონი წარმოადგენს სიმბოლოთა მიმდევრობას, მაგალითად ”ეს PHP კურსია”.

სტრიქონული ცვლადი შეიძლება იყოს ნებისმიერი სახის ტექსტი, რომელიც ორმაგ ან ერთმაგ ბრჭყალებშია მოთავსებული.

## PHP მასივები

მასივი ინახავს სხვადასხვა მნიშვნელობას ერთი სახელის მქონე ცვლადში. ამ მაგალითში ჩვენ შევქმენით მასივები, და შემდეგ გამოვიყენეთ PHP ფუნქცია var\_dump (), რათა შემდეგ გამოვიტანოთ (დავბეჭდოთ) მასივის მონაცემების ტიპი და მისი მნიშვნელობები, ინდექსები იწყება ნულიდან, თუმცა არის შესაძლებლობა ამ პოზიციის შეცვლისა:

ლისტინგი 15 მასივები

<!DOCTYPE html>

<html>

<body>

<?php

$cars=array("Volvo","BMW","Toyota","კოლხიდა");

$natur=array(1,2,3,4,5,6,"ნატურალური რიცხვები");

$namdvili=array(2.5,3.3,4.5,"ნამდვილი რიცხვები");

var\_dump($cars);echo "<br>";

var\_dump($natur);echo "<br>";

var\_dump($namdvili);echo "<br>";

?>

</body>

</html>

ლისტინგ 15-ის შესრულებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_11\_masivi/ მივიღებთ

array(4) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" [3]=> string(21) "კოლხიდა" }   
array(7) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(4) [4]=> int(5) [5]=> int(6) [6]=> string(55) "ნატურალური რიცხვები" }   
array(4) { [0]=> float(2.5) [1]=> float(3.3) [2]=> float(4.5) [3]=> string(49) "ნამდვილი რიცხვები" }

**ტიპი ობიექტი**

ობიექტი არის ისეთი ტიპის მონაცემი, რომელიც ინახავს თავად მონაცემებს და იმას თუ როგორ უნდა დამუშავდეს ეს მონაცემი. ობიექტი უნდა იქნეს ცხადად გამოცხადებული.

უპირველეს ყოვლისა უნდა გამოვაცხადოთ ობიექტის კლასი. ამისთვის უნდა გამოვიყენოთ კლასის სამოსამსახურე სიტყვა. კლასი არის სტრუქტურა, რომელიც შეიცავს თვისებებს და მეთოდებს.

ამის შემდეგ განვსაზღვრავთ მონაცემის ტიპს ობიექტის კლასში, ხოლო შემდეგ გამოიყენება მონაცემის ტიპი კლასის ტიპში. მაგალითად

ლისტინგი 16 ობიექტის ტიპი

<!DOCTYPE html>

<html>

<body>

<?php

class Car

{

var $color;

function Car($color="green") {

$this->color = $color;

}

function what\_color() {

return $this->color;

}

}

function print\_vars($obj) {

foreach (get\_object\_vars($obj) as $prop => $val) {

echo "\t$prop = $val\n";

}

}

// instantiate one object

$herbie = new Car("white");

// show herbie properties

echo "\herbie: Properties\n";

print\_vars($herbie);

?>

</body>

</html>

ლისტინგი 16 -ის გაშვებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_12\_obieqti/ მივიღებთ

**\herbie: Properties color = white**

## NULL მნიშვნელობა

სპეციალური NULL მნიშვნელობა აჩვენებს იმას, რომ ცვლადს არ აქვს მნიშვნელობა. NULL არის ერთადერთი შესაძლო მნიშვნელობა NULL ტიპის მონაცემებისა. NULL მნიშვნელობა განსაზღვრავს არის თუ არა ცვლადი ცარიელი. ასევე სასარგებლოა განასხვაოთ - სტრიქონი ცარიელია და მონაცემთა ბაზაში არის null მნიშვნელობა.

ცვლადები შეიძლება ”დაიცალოს” თუ კი ცვლადს მივანიჭებთ NULL -ს:

ლისტინგი 17 ნული

<!DOCTYPE html>

<html>

<body>

<?php

var\_dump($x);

echo "<br>";

$x="Hello world!";

var\_dump($x);

echo "<br>";

$x=null;

var\_dump($x);

?>

</body>

</html>

ლისტინგი 17 -ის გაშვებისას http://www.kereseli.besaba.com/PHP/laboratoriuli\_5/lab\_5\_13\_null/ მივიღებთ

**NULL   
string(12) "Hello world!"   
NULL**