**ლექცია № 4. PHP სინტაქსი; ცვლადები, მათი ტიპები**

**PHP სინტაქსი**

როგორც ვიცით PHP სკრიპტი შესრულდება სერვერზე, რომლის შედეგი იქნება უბრალო HTML და რომელიც დააბრუნდება ბრაუზერს.

PHP სკრიპტის შეიძლება განთავსდეს დოკუმენტში ნებისმიერ ადგილას. PHP სკრიპტი იწყება **<? php** -ით და მთავრდება **?>**, ასევე შესაძლებელია ამ ტეგების გამოყენება **php** -ის გარეშე და ტეგ <script> -ში language ატრიბუტისთვის **php** -ს მინიჭებით.

PHP ფაილის გაფართოება ჩუმათობის პრინციპით არის ".php".

ჩვეულებრივ PHP ფაილი შეიცავს HTML ტეგებს და ზოგიერთი PHP სკრიპტებს.

ამის მაგალითები უკვე განვიხილეთ წინა ლექციის (ლექცია 1-2) მე-8-ე ლისტინგში.

PHP ოპერატორები სრულდება წერტილ-მძიმით (;). ი PHP-კოდის ბლოკის დახურვის ტეგი ასევე ავტომატურად გულისხმობს წერტილ-მძიმეს, ასე რომ თქვენ შეგიძლიათ არ დაასრულოთ წერტილ-მძიმით PHP ბლოკის ბოლო ოპერატორი.

## კომენტარები PHP -ში

კომენტარი PHP-კოდში არის ბწკარი, რომელიც არ იკითხება /არ შესრულებული პროგრამის მიერ. კომენტარების ერთადერთი მიზანი არის ის, რომ იდი უნდა წაიკითხოს იმან ვისაც სურს კოდის რედაქტირება.

კომენტარები სასარგებლო:

* კომენტარები აძლევს საშუალებას სხვა პროგრამისტს გაიგოს თუ რას აკეთებთ - ანუ მას შეუძლია გაიგოს რას ასრულებს კოდის ესა-თუ ის ნაწილი ნაბიჯ-ნაბიჯ.
* შეგახსენებთ თქვენ, თუ რა გააკეთეთ - ზოგჯერ პროგრამისტი უბრუნდება საკუთარ სამუშაოს ერთი ან ორი წლის შემდეგ და ამ შემთხვევაში მას სჭირდება ხელახლა გაერკვეს იმ კოდში, რომელიც გააკეთეშექმნა. კომენტარი შეუძლია შეგახსენოთ, რას ფიქრობდით მაშინ, როცა წერდით კოდს

PHP მხარს უჭერს სამი სახის კომენტარებს, მაგალითად:

ლისტინგი 1

<!DOCTYPE html>

<html>

<body>

**<?php**

**//**ე**ს არის ერთ სტრიქონში მოთავსებული კომენტარი**

**#ესეც ერთ სტრიქონში მოთავსებული კომენტარია**

**/\***

**ეს არის მრავალსტრიქონიანი კომენტარების ბლოკი**

**რომელიც შეიცავს ერთზე მეტ**

**სტრიქონს**

**\*/**

echo "შემოდგომის ყვავილებს დიდებასთან მივიტან!";

**?>**

</body>

</html>

ლისტინგ 1-ის სერვერზე შესრულებისას (გადავდივართ ლინკზე -http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_1\_koment/ ) ვღებულობთ HTML -კოდს:

ლისტინგი №2

<!DOCTYPE html>

<html>

<body>

შემოდგომის ყვავილებს დიდებასთან მივიტან!

</body>

</html>

როგორც ვხედავთ მივიღეთ მხოლოდ HTML-ს ტეგებს, კომენტარებიც კი აღარ არის და ბრაუზერის მეშვეობით ამ HTML-კოდის რეალიზაციით ვღებულობთ -

**შემოდგომის ყვავილებს, დიდებასთან მივიტან!**

**PHP-ს მგძნობელობა რეგისტრის მიმართ**

PHP-ში მომხმარებლის მიერ განსაზღვრული ყველა ფუნქცია, კლასი და სამოსამსახურე სიტყვები არ არიან მგრძნობიარე რეგისტრის მიმართ.

 ქვემოთ მოყვანილ მაგალითში, სამივე წარმოდგენა echo-სი არის კანონიერი (და ექვივალენტური):

ლისტინგი 3.

<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>

ლისტინგი 3 -ს გამოძახების შედეგად (ლოკალური სერვერის შემთხვევაში ბრაუზერის სამისამართო სტრიქონში ავკრიფოთ - http://localhost/leqcia\_4/mgZnobeloba\_brZanebis/ ; ხოლო ინტერნერ სივრცეში დასორებული სერვერისთვის - http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_1\_koment/ მივიღებთ)

Hello World!
Hello World!
Hello World!

მაგრამ PHP ში რეგისტრის მიმართ მგძნობელობა ცვლადების მიმართ არსებითია.

ქვემოთ მოყვანილ მაგალითში, მხოლოდ პირველი გამოცხადება (echo-სი) მოგვცემს დისპლეიზე $color ცვლადი მნიშვნელობის გამოტანას (ეს იმიტომ, რომ $color, $ COLOR და $coLOR ცვლადები განიხილება, როგორც სამი სხვადასხვა ცვლადი):

ლისტინგი №4

<!DOCTYPE html>

<html>

<body>

<?php

$color="წითელი";

echo "ჩემი მანქანის ფერია " . $color . "<br>";

echo "ჩემი სახლის ფერია " . $COLOR . "<br>";

echo "ჩემი ნავის ფერია " . $coLOR . "<br>";

?>

</body>

</html>

ლისტინგ 4-ის სერვერზე შესრულებისას (გადავდივართ ლინკზე - <http://kereseli.besaba.com/PHP/laboratoriuli_4/lab_4_3_cvladebis_mgrZnobeloba_registri/>) შედეგად მივიღებთ:

ჩემი მანქანის ფერია წითელი
ჩემი სახლის ფერია
ჩემი ნავის ფერია

**ცვლადები PHP-ში**

ცვლადები წარმოადგენენ ”კონტეინერებს” მონაცემთა შესანახად და ამ კონტეინერებს მათთვის ინდივიდუალური სახელები, იდენტიფიკატორები გააჩნიათ:

მაგალითად - ლისტინგი 5 - lab\_4\_4\_cvladebi\_konteineri

<!DOCTYPE html>

<html>

<body>

<?php

$x=5;//$x კონტეინერში მოთავსებულია მონაცენი "5"

$y=6;//$y კონტეინერში მოთავსებულია მონაცენი "6"

$z=$x+$y;/\* $z კონტეინერში მოთავსებულია $x და $y

კონტეინერებში არსებული მონაცემების ჯამი (5+6) "11"

\*/

echo $z;

?>

</body>

</html>

ლისტინგი 5-ის შესრულებისას (<http://kereseli.besaba.com/PHP/laboratoriuli_4/lab_4_4_cvladebi_konteineri/>) მივიღებთ

**11**

PHP-ში ისევე, როგორც ალგებრაში, სადაც სიმბოლო - ასოს (მაგ. x-ს) მნიშვნელობაა 5, გამოსახულებისთვის z=x+y, ჩვენ შეგვიძლია გამოვთვალოთ z-ის მნიშვნელობა - 11. PHP -ში ალგებრაში მიღებულ ასოებს -სიმბოლოებს **ცვლადები** ეწოდება.

ისევე, როგორც ალგებრაში ცვლადებს PHP-ში შეიძლება მინიჭებული ჰქონდეთ მნიშვნელობა, მაგ. x=5; ან გამოსახულება z=x+y . ცვლადებს შეიძლება ჰქონდეთ მოკლე სახელები - იდენტიფიკატორები ან უფრო თვალსაჩინო, შინაარსიანი სახელი (მაგალითად asaki, manqanissaxeli, mTeli\_mniSvneloba, name და ა.შ.)

არსებობს შემდეგი წესები PHP-ს ცვლადების სახელებისთვის:

* ცვლადის სახელი უნდა იწყებოდეს სიმბოლოთი - $, რომელსაც მოსდევს ცვლადის არსებითი სახელი;
* ცვლადის არსებითი სახელი უნდა იწყებოდეს ამ ასოთი ან ქვედა ხაზის სიმბოლოთი;
* ცვლადის არსებითი სახელი არ უნდა იწყებოდეს რიცხვით;
* ცვლადის არსებითი სახელი შეიცავს ასოით-ციფრულ ნიშნებს და ქვედა ხაზგასმის სიმბოლოს (a-z, 0-9 და \_), ამასთან ასოები შეიძლება გამოყენებულ იქნენ, როგორც დაბალ, ასევე მაღალ რეგისტრში;
* ცვლადის არსებითი სახელი მგრძნობიარეა რეგისტრის მიმართ, მაგალითად ცვლადები $q და $Q ორი განსხვავებული ცვლადია!!!

**ცვლადების გამოცხადება PHP-ში**

PHP-ს არ გააჩნია ბრძანება ცვლადების გამოცხადებისთვის. ცვლადი ითვლება გამოცხადებულად მას შემდეგ, რაც მას პირველად მიენიჭება მნიშვნელობა: მაგალითად

ლისტინგი 6

<!DOCTYPE html>
<html>
<body>

<?php
$txt="დილა მშვიდობისა საქართველო!";
$x=5;
$y=10.5;

echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>

</body>
</html>

ლისტინგი 6-ის შესრულებისას (http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_5\_cvladebis\_gamocxadeba/) ვღებულობთ:

**დილა მშვიდობისა საქართველო!
5
10.5**

ლისტინგი 6-ის ჩატვირთვისას ცვლადს txt მიენიჭება მნიშვნელობა დილა მშვიდობისა საქართველო! ; x ცვლადს 5; ხოლო y ცვლადს 10.5. ეს ჩვლადები ამის შენდეგ ითვლება გამოცხადებულად!

შევნიშნოთ, რომ თუ კი ცვლადს ვანიჭებთ ტექსტურ მნიშვნელობას, მაშინ ეს ტექსტი უნდა იქნას მოთავსებული ბრჭყალებში. ბრჭყალები შეიძლება იყოს როგორც ერთმაგი, ასევე ორმაგი, მაგალითად

$str='სტრიქონთა რაოდენობა ცხრილში';

$str1="სვეტების რაოდენობა ცხრილში";

მაგრამ არსებობს ორმაგი ბრჭყალების გამოყენების ნიუანსი, ამ შემთხვევაში (როცა მოცემულია ორმაგი ბრჭყალები) PHP-ს კომპილატორი ასეთ ცვლადში (შიგნით - ინტრო) ეძებს კიდევ რომელიღაც სხვა ცვლადს, ერთმაგი ბრჭყალების შემთხვევაში ეს ასე არაა. განვიხილოთ მაგალითი

ლისტინგი 7

<!DOCTYPE html>
<html>
<body>

<?php
$ricxvi=5;

$str='სტრიქონთა რაოდენობა ცხრილში';

$str1="სვეტების რაოდენობაზე ცხრილში";

$str2="$str მეტია $str1 $ricxvi -ით";

echo $str2;
?>
</body>
</html>

ლისტინგი 7-ის შესრულებისას

(<http://kereseli.besaba.com/PHP/laboratoriuli_4/lab_4_6_brWyalebi/>) ვღებულობთ:

**სტრიქონთა რაოდენობა ცხრილში მეტია სვეტების რაოდენობაზე ცხრილში 5 -ით**.

მაგალითი, ლისტინგ 6-ში აჩვენებს, რომ PHP ენა არის სუსტად ტიპიზირებული, რადგანაც ჩვენ არ გვჭირდება ცვლადის ტიპის გამოცხადება. იმას, თუ ცვლადი რა ტიპის უნდა იყოს PHP ავტომატურად განსაზღვრავს, როცა ამ ცვლადს მნიშვნელობა მიენიჭება. ზოგ სხვა ენაში ეს ასე არაა, მაგალითად ენებში C,C++, Java , პროგრამისტმა წინასწარ უნდა გამოაცხადოს ესა თუ ის ცვლადი რა ტიპისაა.

 **PHP ცვლადების მოქმედების არე**

PHP-ში ცვლადების დეკლარირება -გამოცხადება, ანუ მისთვის მნიშვნელობის მინიჭება შესაძლებელია სკრიპტის ნებისმიერ ადგილას!

ცვლადის მოქმედების არე - ეს არის სკრიპტის ის ნაწილი რომელშიც შესაძლებელია ამ ცვლადზე მითითება ან მისი გამოყენება.

PHP-ში არის ცვლადების მოქმედების სამი არე:

* ლოკალური;
* გლობალური;
* სტატიკური.

ცვლადი, რომელიც გამოცხადებულია *ფუნქციის* **გარეთ**, გააჩნია გლობალური მოქმედების არე და შეიძლება გამოყენებულ იქნეს მხოლოდ *ფუნქციი*ს **გარეთ**.

ცვლადი, რომელიც გამოცხადებულია *ფუნქციის* **შიგნით**, გააჩნია ლოკალური მოქმედების არე და შეიძლება გამოყენებულ იქნეს მხოლოდ *ფუნქციის* **შიგნით**.

ქვემოთ მოყვანილი მაგალითი აჩვენებს ლოკალურ და გლობალურ ცვლადების მოქმედების არეებს:

ლისტინგი 8 - ლოკალური და გლობალური ცვლადების ტესტი

<!DOCTYPE html>

<html>

<body>

<?php

$x=5; // გლობალური მოქმედების არე

function myTest() {

 $y=10; // ლოკალური მოქმედების არე

 echo "<p> ფუნქციის შიგნით ცვლადის ტესტი:</p>";

 echo "ცვლადი x არის: $x";

 echo "<br>";

 echo "ცვლადი y არის: $y";

}

myTest();

echo "<p>ფუნქციის გარეთ ცვლადის ტესტი:</p>";

echo "ცვლადი x არის: $x";

echo "<br>";

echo "ცვლადი y არის: $y";

?>

</body>

</html>

ლისტინგი 8-ის გაშვების შემდეგ (http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_7\_globaluri\_lokaluri/) ვღებულობთ:

**ფუნქციის შიგნით ცვლადის ტესტი:**

**ცვლადი x არის:
ცვლადი y არის: 10**

**ფუნქციის გარეთ ცვლადის ტესტი:**

**ცვლადი x არის: 5
ცვლადი y არის:**

ზემოთ მოყვანილ მაგალითში არსებობს ორი ცვლადის $ x და $ y და ფუნქცია myTest (). $ x გლობალური ცვლადია, რადგან ის გაცხადებულია ფუნქციის გარეთ, ხოლო $ y არის ლოკალური ცვლადი, რადგან ისგამოცხადებულია ფუნქციის შიგნით.

როდესაც ჩვენ გამოგვაქვს ორი ცვლადის მნიშვნელობა myTest () ფუნქციის შიგნით, მაშინ იბეჭდება $ y-ის მნიშვნელობა, რადგანაც ის გამოცხადებულია როგორც ლოკალური ცვლადი, მაგრამ ვერ ბეჭდავს $ x-ის მნიშვნელობას, რადგანაც იგი გამოცხადდა ფუნქციის გარეთ.

მაშინ, როდესაც ჩვენ გამოგვაქვს ორი ცვლადის მნიშვნელობა myTest () ფუნქციის გარეთ, იბეჭდება $ x ცვლადის მნიშვნელობა, მაგრამ ვერ იბეჭდება $ y ცვლადის მნიშვნელობა, რადგან ეს უკანასკნელი ლოკალური ცვლადია, ის myTest () ფუნქციის შიგნითაა გამოცხადებული.

შევნიშნოთ, რომ ჩვენ შეგვიძლია გვქონდეს ერთი და იგივე ცვლადი სხვადასხვა ფუნქციაში, და ეს არ შექმნის პრობლემას, რადგანაც ლოკალური ცვლადები ყოველი კონკრეტული ფუნქციის შიგნით არიან აღქმადი.

ისმის კითხვა, როგორ გამოვიყენოთ ფუნქციის შიგნით გლობალური ცვლადი, რადგანაც ეს ხშირად ხდება საჭირო და პირიქით, ფუნქციის მიერ გამოთვლილი ლოკალური ცვლადის მნიშვნელობა , როგორ უნდა ვაქციოთ გლობალურ ცვლადად, რათა ის გამოვიყენოთ ფუნქციის გარეთ?

პირველ კითხვაზე ორი პასუხია, რომლის საფუძველია მოსამსახურე (დარეზერვირებული) სიტყვა:

1. მოსამსახურე სიტყვა **global** გამოიყენება გლობალური ცვლადის ფუნქციის შიგნით გამოსაყენებლად, ამისათვის მოსამსახურე სიტყვა **global** უნდა იქნეს ფუნქციის შიგნით მოთავსებული ცვლადების წინ, იმ ცვლადებისა, რომლებიც არიან გლობალური:

ლისტინგი 9 - გლობალური ცვლადის ლოკალურად გამოყენება

<!DOCTYPE html>

<html>

<body>

<?php

$x=5;

$y=10;

function myTest() {

 global $x,$y;

 $y=$x+$y;

}

myTest(); // ფუნქციის გაშვება

echo $y; // $y ცვლადის ახალი მნიშვნელობის გამოტანა

?>

</body>

</html>

ლისტინგ 9-ის გაშვების შემდეგ (http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_8\_globaluri\_lokalurSi/ ) მივიღებთ **15**-ს

1. PHP ინახავს ყველა გლობალურ ცვლადს ერთგანზომილებიანი მასივის სახით $GLOBALS[ინდექსი]. ინდექსად აღებულია ცვლადის სახელი. და ამ მასივზე წვდომა შესაძლებელის ფუნქციის შიგნითანაც, ამიტომაც შესაძლებელია ფუნქციაში გლობალური ცვლადი გამოყენებულ იქნეს ამ მასივის ელემენტის საშუალებით. მაგალითი:

ლისტინგი 10 - გლობალური ცვლადის ლოკალურად გამოყენება მასივის საშუალებით

lab\_4\_9\_globaluri\_lokalurSi-masiviT

<!DOCTYPE html>

<html>

<body>

<?php

$x=5;

$y=10;

function myTest() {

 $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];

}

myTest();

echo $y;

?>

</body>

</html>

ლისტინგ 10-ის გაშვების შემდეგ (http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_9\_globaluri\_lokalurSi\_masiviT/) მივიღებთ ისევ **15**-ს

როგორ უნდა ვაქციოთ ლოკალური ცვლადი გლობალურ ცვლადად, რათა ის გამოვიყენოთ ფუნქციის გარეთ?

ამისთვის გამოვიყენოთ მოსამსახურე სიტყვა **static -** რადგანაც ფუნქციის შესრულების შემდეგ ლოკალური ცვლადების მნიშვნელობები იკარდება - წაიშლება, იმისთვის, რომ ჩვენთვის საჭირო ლოკალური ცვლადის მნიშვნელობა შევინარჩუნოთ, მას წინ უნდა დავუწეროთ მოსამსახურე სიტყვა **static** , და მოვათავსოთ ის პირველ გამოცხადებაში, მაგალითად:

ლისტინგი 11 - ლოკალური ცვლადის გლობალურად გამოყენება

<!DOCTYPE html>
<html>
<body>

<?php
function myTest() {
   static $x=0;
   echo $x;
   $x++;
}

myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
?>

</body>
</html>

ლისტინგ 11-ის გაშვების (http://kereseli.besaba.com/PHP/laboratoriuli\_4/lab\_4\_10\_lokaluri\_globalurSi/) შემდეგ გვაქვს:

**0
1
2
3
4**

ამ შემთხვევაში, როცა ეს ფუნქცია **static** -ით რამდენჯერმე იქნება გამოძახებული, ყოველი გამოძახებისას, ამ ლოკალურ ცვლადში იქნება ის მნიშვნელობა რომელიც შეიქმნა - მოთავსთა მასში, წინა გამოძახებისას - როცა შესრულდა ეს ფუნქცია.