**ლექცია N 7 დინამიური ცვლადები**

დაპროგრამების ყველა ენაში შესაძლებელია ცვლადების მნიშვნელობის შეცვლა; უფრო მცი­რე რაოდენობის დაპროგრამების ენაში შესაძლებელია ცვლადების ტიპის შეცვლა და მხო­ლოდ ერთეულ დაპროგრამების ენებშია შესაძლებელი, რომ ცვლადს სახელი შეეცვალოს - PHP ამის საშუალებას იძლევა. მათ ცვლადების ცვლადები ეწოდება და იძლევა საშუალებას დინამიურად ვცვალოთ ცვლადების სახელები. ასეთი ტიპის ცვალდებს დინამიურ ცვლა­დებსაც უწოდებენ. ცვლადის სახელის შეეცვალის საფუძველს წარმოადგენს იდეა იმის შესახებ, რომ გამოყენებულ იქნეს ერთი რომელიმე ცვლადის მნიშვნელობა მეორე ცვლადის სახელად, მაგალითად

*ლისტინგი 1*

<!DOCTYPE html>

<html>

<body>

<?php

$str2="saburavi\_raodenoba"; // ვქმნით სტრიქონულ ცვლადს

$$str2=5; //$saburavi\_raodenoba ცვლადის მაგიერ გამოიყენება ცვლადი $$str2

echo 'ცვლადი $$str2 არსებობს და მისი ტიპი და მნიშვნელობაა = ';

echo var\_dump($$str2);

echo "<br>$$str2=${$str2}<br>";

echo "ცვლადი \$saburavi\_raodenoba არსებობს და მისი მნიშვნელობაა = $saburavi\_raodenoba"; /\* დემონსტრაცია იმისა, რომ ცვლადები

$$str2 და $saburavi\_raodenoba ერთი და იგივე მნიშვნელობას იღებენ \*/

?>

</body>

</html>

ლისტინგი 1-ის გაშვებისას ვღებულობთ:

**ცვლადი $$str2 არსებობს და მისი ტიპი და მნიშვნელობაა = int(5)   
$saburavi\_raodenoba=5  
ცვლადი $saburavi\_raodenoba არსებობს და მისი მნიშვნელობაა = 5**

მივაქციოთ ყურადღება იმას, რომ გამოტანის კონსტრუქცია echo-ში თუ კი იქნება ჩასმული $$str2 - ორმაგი $-ით, მაშინ მონიტორზე გამოდის **$saburavi\_raodenoba**, ხოლო თუ კი ორმაგ დოლარის მნიშვნელობაში პირველი დოლარის მნიშვნელობის შემდეგ არსებულ გამოსახულებას ჩავსვამთ ფიგურულ ფჩხილში, გამოიტანება მისი მნიშვნელობა, ჩვენს შემთხვევაში **5.**

ერთი რომელიმე ცვლადის მნიშვნელობა მეორე ცვლადის სახელის ნაწილად შეიძლება იქნეს გამოყენებული, მაგალითად:

*ლისტინგი 2*

!DOCTYPE html>

<html>

<body>

<?php

$id\_menu = 3;

$str = "active$id\_menu"; // "active3"

$$str = 1; // $active3 = 1;

if(isset($active3))

echo "ცვლადი \${$str} არსებობს და ტოლია $active3";

?>

</body>

</html>

ლისტინგი 2-ის გაშვებისას ვღებულობთ:

**ცვლადი $active3 არსებობს და ტოლია 1**

ამდენად, მნიშვნელობა ცვლადისა, რომელიც პირველი $ნიშნის შემდეგ მოდის, აღიქმება, როგორც ახალი ცვალდის სახელი. ამის გამო, თუ კი სტრინგ $str -ს აქვს მნიშვნელობა ”active3”, მაშინ ახალი ცვლადის სახელი იქნება $active3

დინამიური ცვლადის შექმნა შესაძლებელია eval() ფუნქციის საშუალებით. ეს ფუნქცია პარამეტრად ღებულობს PHP-კოდს და ასრულებს მას, მაგალითად

*ლისტინგ 3* დინამიური ცვლადის შექმნა eval() ფუნქციით

<!DOCTYPE html>

<html>

<body>

<?php

$code = '$str = "ეს ფოკუსია!<br>";

echo $str;';

eval($code);

echo $str;

?>

</body>

</html>

ლისტინგი 3-ის გაშვებისას ვღებულობთ:

**ეს ფოკუსია!  
ეს ფოკუსია!**

ამ ლისტინგში არსებითია ის, რომ ცვლადი $str, განსაზღვრული სტრინგში $code, მიწვდომადია არა მხოლოდ ამ სტრინგ-სტრიქონის შიგნით, არამედ ფუნქცია eval -ის შესრულების შემდეგაც. ამ ეფექტზეა დამყარებული დინამიური ცვლადების შექმნა, როგორც ცვლადების მნიშვნელობის დინამიურ ცვლილება, ასევე ცვლადების სახელების დინამიურ ცვლილებას, მაგალითად რვაობითი რიცხვების, რომელიც შეიძლება წვდომის უფლებას განსაზღვრავს:

*ლისტინგი 4*

<!DOCTYPE html>

<html>

<body>

<?php

$number = 755;

eval("\$mode = 0$number;");

if(isset($mode)) echo $mode; // 493

echo '<br>';

$id\_menu = 3;

eval("\$active$id\_menu = 1;");

if(isset($active3))

echo "ცვლადი \$active3 არსებობს და ტოლია $active3 -ის";

?>

</body>

</html>

ლისტინგი 4-ის გაშვებისას ვღებულობთ:

**493  
ცვლადი $active3 არსებობს და ტოლია 1 -ის**

**შეკვეთების ფორმა**

ვთქვათ ავტონაწილების მაღაზია, რომლის მფლობელია ვინმე ბუბა, ვაჭრობს ინტერნეტის საშუალებით და ვებ-პროჰრამისტებმა შეუქმნეს მას შეკვეთების ფორმა html კოდში და მისი დამუშავებისთვის PHP კოდი. ორივე ვარიანტო მოცემულია ლისტინგ 5 და 6-ის საშუალებით

*ლისტინგი 5* შეკვეთის ფორმა

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />

<title>ავტონაწილები ბუბასგან</title>

</head>

<body>

<form action="processorder.php" method="post">

<table border="0">

<tr bgcolor="#cccccc">

<td width="160">საქონელი</td>

<td width="15">რაოდენობა</td>

</tr>

<tr>

<td>საბურავები</td>

<td align="center"><input type="text" name="tireqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>ზეთი</td>

<td align= "center"><input type="text" name="oilqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>აალების სანთლები</td>

<td align="center"><input type="text" name="sparkqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>როგორ გახვედით ბუბაზე?</td>

<td>

<select name="find">

<option value = "a">მე მუდმივი კლიენტი ვარ</option>

<option value = "b">სატელევიზიო რეკლამით</option>

<option value = "c">სატელეფონო ცნობარში</option>

<option value = "d">ზეპირი რეკომენდაცია</option>

</select>

</td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit"

value="გააგზავნეთ შეკვეთა" /></td>

</tr>

</table>

</form>

</body>

</html>

*ლისტინგი 6* შეკვეთის დამუშავება

<html>

<head>

<title>ავტონაწილები ბუბასგან — შეკვეთის შედეგები</title>

</head>

<?php

// ცვლადების მოკლე სახელების შექმნა

$tireqty = $\_POST['tireqty'];

$oilqty = $\_POST['oilqty'];

$sparkqty = $\_POST['sparkqty'];

?>

<body>

<h1>ავტონაწილები ბუბასგან</h1>

<h2>შეკვეთის შედეგი</h2>

<?php

echo "<p>შეკვეთა დამუშავდა".date('H:i, jS F Y')."</p>";

// echo '<p>შეკვეთილია: </p>';

// echo $tireqty.' საბურავი<br />';

// echo $oilqty.' ბოთლი ზეთი<br />';

// echo $sparkqty.' აალების სანთლელი<br />';

$totalqty = 0;

$totalqty = $tireqty + $oilqty + $sparkqty;

echo "შეკვეთილი საქონელი: ".$totalqty."<br />";

$totalamount = 0.00;

define('TIREPRICE', 100);

define('OILPRICE', 10);

define('SPARKPRICE', 4);

$totalamount = $tireqty \* TIREPRICE

+ $oilqty \* OILPRICE

+ $sparkqty \* SPARKPRICE;

echo "ჯამში: $".number\_format($totalamount,2)."<br />";

$taxrate = 0.10; // ადგილობრივი მოსაკრებელი გაყიდვებისგან შეადგენს 10%

$totalamount = $totalamount \* (1 + $taxrate);

echo "მთლიანად, გაყიდვებისგან ბაჟის ჩართვით: $".number\_format($totalamount,2)."<br />";

?>

</body>

</html>

ჩვენ ნაწილ - ნაწილ მივიდეთ ამ კონდიციამდე. შევქმნათ ფორმების ძირითადი ნაწილი

*ლისტინგ 7*-ში

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />

<title>ავტონაწილები ბუბასგან</title>

</head>

<body>

<form action="processorder\_1.php" method="post">

<table border="0">

<tr bgcolor="#cccccc">

<td width="160">საქონელი</td>

<td width="15">რაოდენობა</td>

</tr>

<tr>

<td>საბურავები</td>

<td align="center"><input type="text" name="tireqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>ზეთი</td>

<td align= "center"><input type="text" name="oilqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>აალების სანთლები</td>

<td align="center"><input type="text" name="sparkqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit"

value="გააგზავნეთ შეკვეთა" /></td>

</tr>

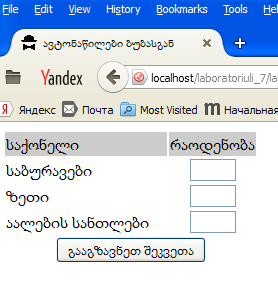
</table>

</form>

</body>

</html>

ლისტინგ 7-ის გაშვებისას მივიღებთ:



ლისტინგ 7-ის მე-7-ე სტრიქონში ფორმის ქმედებას (action) მინიჭებული აქვს PHP-სცენარის სახელი (processorder\_1). ზოგადად action ატრიბუტის მნიშვნელობა ეს არის URL-მისამართი, რომელიჩ ჩაიტვირთება მას შემდეგ რაც დააჭერთ ღილაკს ”გააგზავნეთ შეკვეთა” ( Submit). ის მონაცემები, რომელიც მომხმარებელმა შეიყვანა ფორმაში გადაიგზავნება ამ URL-მისამართზე იმ მეთოდის საშუალებით, რომელიც მითითებულია ატრიბუტ method-ში: ეს ან GET-მეთოდია -ამ შემთხვევაში ფორმის მონაცემები დაემატება URL-მისამართის ბოლოს, ან POST- მეთოდია -მონაცემები გადაიგზავნება ცალკე შეტყობინების სახით.

გარდა ამისა მივაქციოთ ყურადღება იმას, რომ ფორმის ველებს აქვთ მინიჭებული სახელები, რომლებიც აზრობრივ დატვირთვას ატარებენ აღსაწერი პროცესის მიმართებაში, მაგალითად tireqty - ინგლისურად საბურავების რაოდენობას აღნიშნავს;

Oilqty - ზეთის ბოთლების რაოდენობას;

Sparkqty- აალების სანთლების რაოდენობას.

**ფორმის დამუშავება**

როგორც ვიცით ლისტინგ 7-ში არსებული ფორმრბის დამუშავება უნდა მოხდეს processorder\_1.php -ის მიერ. სწორედ ეს კოდი დავწეროთ

*ლისტინგ 8*-ში:

<html>

<head>

<title>ავტონაწილები ბუბასგან — შეკვეთის შედეგები</title>

</head>

<body>

<h1>ავტონაწილები ბუბასგან</h1>

<h2>შეკვეთის შედეგი</h2>

<?php

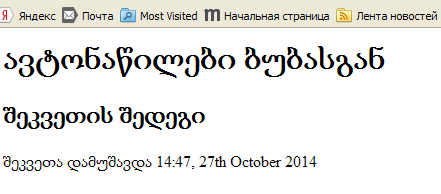
echo "<p>შეკვეთა დამუშავდა".date('H:i, jS F Y')."</p>";

?>

</body>

</html>

ლისტინგ 8-ის გაშვებისას მივიღებთ



ლისტინგ 8-ში ჩართულია ფუნქცია data() – რომლის ფუნქციონირების შემთხვევაში დაფიქ­სიედება processorder\_1.php -სადმი მიმართვის დრო, მაგრამ გამოტანის ოპერატორში არსე­ბული სტრინგით აზრობრივად, საქმე ეხევა შეკვეთის დამშავების დროს. რეალურად ამ კოდ­ში ფორმის მონაცემთა დამუშავება არ ხდება, ჩვენ ჯერ პირველ მიახლოებაში ვიმყოფებით იმ მიზანთან - რასაც მონაცემთა რეალური დამუშავება ჰქვია.

ცოტა მეტი ინფორმაცია მოვიტანოთ data() ფუნქციაზე, მის ფრჩხილებში უნდა იქნეს მითი­თებული არგუმენტი სტრინგის ფორმით, რომელიც განსაზღვრავს, თუ რა ფორმით იქნება დროის მიღება. ჩვენს კონკრეტულ შემთხვევაში H მიუთითებს იმას, რომ საათის ჩვენება იყოს 24 საათიანის ფორმატში, i - წუთები, საჭიროების შემთხვევაში ნულის მითითებოთ - იმისთვის, რომ წუთი იყოს ორნიშნა რიცხვის სახით გამოსახული, j - თვის რიცხვი წინა ნულის გარეშე, S ჩვეულებრივი სუფიქსია, ამ შემთხვევაში “th”, F -თვის სრული დასახელება ინგლისურად, Y - წელი. არსებობს საშუალება, დრო გამოვიტანოთ სხვადასხვა ფორმატით (სტუდენტებმა პრეზენტაციის სახით დაამუშავონ ეს საკითხი)

**წვდომა ფორმის ცვლადებზე**

ამრიგად, როდესაც ფორმის ველები შევსებულია და გადაგზავნილია სერვრტზე, სკრიპტმა უნდა მოახდინოს მისი დამუშავება. ფორმის ველებში ჩაწერილ და მერე გადაგზავნილ მონაცემებს, ფორმის ცვლადები ეწოდება, როგორც გვახსოვს, და მათზე წვრომა სკრიპტში ხდება შემდეგნაირად.

თუ ფორმის მონაცემები გაიგზავნა POST მეთოდით, მაშინ ფორმის ცვლადები მოთავსდება მასივში $\_ POST, და თუ კი გვინდა მოვახდინოთ წვდომა ფორმის tireqty ველში მოთავსებულ მნიშვნელობაზე, მაშინ ის უნდა ჩავწეროთ შემდეგნაირად:

$\_ POST[' tireqty '] – წვდომის ასეთ ჩაწერას მოკლე სტილი ეწოდება; მოკლე სტილი ყველაზე უფრო მისაღებია და ერთგვარ სტანდარტს წარმოადგენს PHP-ში.

არსებობს წვდომის სხვა სტილები, რომლებიც ნაკლებად არის რეკომენდებული,ასე მაგალითად:

$tireqty - მოკლე სტილი - ის მოსახერხებელია, მაგრამ მოითხოვს ჩართულ იქნეს registar\_globals -ს კონფიგურაციული ელემენტები, რომელიც უსაფრთხოების მოსაზრების გამო, როგორც წესი გამორთულია;

$HTTP\_POST\_VARS[' tireqty '] - გრძელი სტილი-ეს სტილი მოძველებულია, ის შეიძლება გამოირთოს registar\_long\_arrays -ს კონფიგურაციის დირექტივის საშუალებით და მაშინ იზრდება სკრიპტის წარმადობა.

თუ ფორმის მონაცემები გაიგზავნა GET მეთოდით, მაშინ ფორმის ცვლადები მოთავსდება მასივში $\_GET . ამ შემთხვევაში წვდომა ფორმის tireqty ველში მოთავსებულ მნიშვნელობაზე მოხდება შემდეგნაირად - $\_GET[' tireqty '].

ყველა იმ მონაცემზე რომელიც გაიგზავნა თუ გინდ POST მეთოდით, თუ გინდ GET მეთოდით, წვდომა შესაძლებელია $\_REQUEST მასივის საშუალებით - მაგალითად $\_REQUEST[' tireqty ']

ეხლა ეროცა შევისწავლეთ თუ როგორ მოვახდინოთ წვდომა ფორმის ცვლადებზე ჩავსვათ ის ლისტინგ 8-ში, რომელიც შეკვეთის დამუშავებას ახდენს:

ლისტინგი 9 ფორმის ცვლადების გამოყენება

<html>

<head>

<title>ავტონაწილები ბუბასგან — შეკვეთის შედეგები</title>

</head>

<body>

<h1>ავტონაწილები ბუბასგან</h1>

<h2>შეკვეთის შედეგი</h2>

<?php

echo "<p>შეკვეთა დამუშავდა".date('H:i, jS F Y')."</p>";

// ცვლადების მოკლე სახელების შექმნა

$tireqty = $\_POST['tireqty'];

$oilqty = $\_POST['oilqty'];

$sparkqty = $\_POST['sparkqty'];

echo '<p>შეკვეთილია: </p>';

echo $tireqty.' საბურავი<br />';

echo $oilqty.' ბოთლი ზეთი<br />';

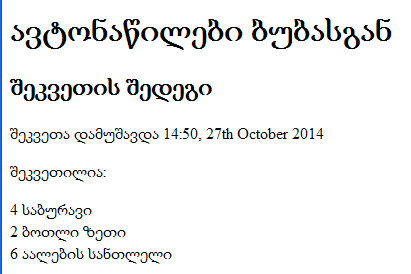
echo $sparkqty.' აალების სანთლელი<br />';

?>

</body>

</html>

ლისტინგი 9-ის გაშვების შემდეგ მივიღებთ



თუ კი ლისტინგ 9-ში დავუმატებთ

echo '<p>შეკვეთილია: </p>';

echo $\_POST['tireqty'].' საბურავი<br />';

echo $\_POST['oilqty'].' ბოთლი ზეთი<br />';

echo $\_POST['sparkqty'].' აალების სანთლელი<br />';

ნაწილს, მაშინ თვალსაჩინო ხდება, რომ შეგვეძლო არ გამოგვეყენებინა ეს

კოდი:

$tireqty = $\_POST['tireqty'];

$oilqty = $\_POST['oilqty'];

$sparkqty = $\_POST['sparkqty'];

და პირდაპირ საშუალო სტილი გამოგვეყენებინა, შედეგს იგივეს მივიღებდით, მართლაც ლისტინგ 9-ს გაშვებით ორჯერ მივიღებთ შეკვეთილი საქონელის რაოდენობის გამოტანას:

# ავტონაწილები ბუბასგან

**შეკვეთის შედეგი**

შეკვეთა დამუშავდა 15:04, 27th October 2014

შეკვეთილია:

4 საბურავი  
2 ბოთლი ზეთი  
6 აალების სანთლელი

შეკვეთილია:

4 საბურავი  
2 ბოთლი ზეთი  
6 აალების სანთლელი

მაგრამ თუ კი ლისტინდი 9-დან პირდაპირ ცვლადების მოკლე სტილზე გადავალთ, მაშინ მივიღებთ, რომ თავდაპირველად არ ხდება შეკვეთილი საქონელის რაოდენობის გამოტანა, რაც იმის მანიშნებელია, რომ, არაა ჩართული registar\_globals -ს კონფიგურაციული ელემენტები, რომელიც უსაფრთხოების მოსაზრების გამო, როგორც წესი გამორთულია

**ავტონაწილები ბუბასგან**

**შეკვეთის შედეგი**

შეკვეთა დამუშავდა 15:06, 27th October 2014

შეკვეთილია:

საბურავი  
ბოთლი ზეთი  
აალების სანთლელი

შეკვეთილია:

4 საბურავი  
2 ბოთლი ზეთი  
6 აალების სანთლელი

ვნახოთ, ეხლა GET მეთოდით ფორმის მონაცემების გადაცემა. ამასთან ჩავრთოთ ახალი ფორმა.

*ლისტინგი 10* შეკვეთის შექმნა GET მეთოდი საშუალებით

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />

<title>ავტონაწილები ბუბასგან</title>

</head>

<body>

<form action="processorder\_2.php" method="get">

<table border="0">

<tr bgcolor="#cccccc">

<td width="160">შემკვეთის გვარი</td>

<td width="80">შემკვეთის სახელი</td>

</tr>

<tr>

<td align="center"><input type="text" name="gvari" size="20"

maxlength="20" /></td>

<td align="center"><input type="text" name="saxeli" size="20"

maxlength="20" /></td>

</tr>

</table>

<table border="0">

<tr bgcolor="#cccccc">

<td width="160">საქონელი</td>

<td width="15">რაოდენობა</td>

</tr>

<tr>

<td>საბურავები</td>

<td align="center"><input type="text" name="tireqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>ზეთი</td>

<td align= "center"><input type="text" name="oilqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td>აალების სანთლები</td>

<td align="center"><input type="text" name="sparkqty" size="3"

maxlength="3" /></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit"

value="გააგზავნეთ შეკვეთა" /></td>

</tr>

</table>

</form>

</body>

</html>

*ლისტინგი 11* GET მეთოდით დამუშავება

<html>

<head>

<title>ავტონაწილები ბუბასგან — შეკვეთის შედეგები</title>

</head>

<body>

<h1>ავტონაწილები ბუბასგან</h1>

<h2>შეკვეთის შედეგი</h2>

<?php

echo "<p>შეკვეთა დამუშავდა ".date('H:i, jS F Y')."</p>";

// ცვლადების მოკლე სახელების შექმნა

$gvari=$\_GET['gvari'];

$saxeli=$\_GET['saxeli'];

echo $saxeli.' '.$gvari.'- შეკვეთის განმახორციელებელი';

$tireqty = $\_GET['tireqty'];

$oilqty = $\_GET['oilqty'];

$sparkqty = $\_GET['sparkqty'];

echo '<p>შეკვეთილია: </p>';

echo $tireqty.' საბურავი<br />';

echo $oilqty.' ბოთლი ზეთი<br />';

echo $sparkqty.' აალების სანთლელი<br />';

?>

</body>

</html>

ლისტინგ 10-ის გაშვებისას მივიღებთ:

**ავტონაწილები ბუბასგან**

**შეკვეთის შედეგი**

შეკვეთა დამუშავდა 22:52, 28th October 2014

კაცია არაბული- შეკვეთის განმახორციელებელი

შეკვეთილია:

4 საბურავი  
8 ბოთლი ზეთი  
6 აალების სანთლელი

$\_REQUEST