

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet

Java 8: Koncepte të Avancuara në PHP (Pointerët dhe SQL Injection)



Pointerët

Referencat në PHP na mundësojnë qasjen e përbajtjes së njëjtë të variables prej emrave të ndryshëm të variablave. Nuk janë analog me pointerat në C, p.sh, nuk mund të aplikojmë aritmetik në ta, pasi referencat në PHP nuk paraqesin adresa të memories.

Me anë të referencave mund të performojmë tri operacione bazike: *shoqërimi nëpërmes referencës*, *përcjellja nëpërmes referencës*, dhe *kthimi nëpërmes referencës*.

Shoqërimi përmes Referencës:

Detyra 1: Dy variabla të cilat tregojnë në të njëjtën përbajtje:

```
<?php
    $a =&$b;
    /*me kete rast dy variablat jane te barabarta,
    te dy variablat tregojne ne te njejten permbajtje
    dmth a nuk pointon ne b apo anasjelltas.*/
    $a=10;
    echo "Vlera e variables b:". $b;
?>
```

Detyra 2: Të tregohet impakti i referencës në vlerën e një variable.

```
<?php
    $ref = 0;
    $row =&$ref;
    echo "Vlerat e anetareve te vektorit<br>";
    //Vlerat e anetareve te vargut
    foreach (array(1, 2, 3) as $row) {
        echo $row."<br>";
    }
    echo "-----<br>";
    echo "Vlera e variables ref:". $ref; // elementi i fundit nga iterimi i vargut?>
```

Detyra 3: Vendosja e referencave në mes të variablave dhe anëtarëve të vargut.

```
<?php
    $a = 1;
    $b = array(2, 3);
    //lidhja e anetareve te vektorit me reference
    $arr = array(&$a, &$b[0], &$b[1]);
    $arr[0]++;
```

```
$arr[1]++;  
$arr[2]++;  
/* $a == 2, $b == array(3, 4); */  
echo "Vlera e variables a:". $a. "<br>";  
echo "Vlera e aneterit te pare te vektorit b:". $b[0] . "<br>";  
echo "Vlera e aneterit te dyte te vektorit b:". $b[1] . "<br>";  
?>
```

Detyra 4:

Jepni një shembull me shoqërime të kombinuara (me dhe pa reference).

```
<?php  
$a = 1;          /* Shoqerim me vlere */  
$b =&$a;         /* Shoqerim me reference */  
$c = $b;         /* Shoqerim me vlere – me kete rast do i percillet  
                  vetem vlere ne te cilen tregon b, jo edhe referencia */  
$c = 7;          /* Pasi $c nuk eshte reference; ku shoqerim nuk do te kete  
                  efekt ne $a apo $b */  
  
echo "Vlera e variables b=" . $b. "<br>";  
echo "Vlera e variables c=" . $c. "<br>";  
  
/* Shoqerimi i array variablave */  
$arr = array(1);  
$a =&$arr[0];    /* $a dhe $arr[0] ndajne te njejten reference */  
$arr2 = $arr;    /* shoqerim me vlere */  
$arr2[0]++;      /* $a == 2, $arr == array(2) */  
/* Edhe pse $arr nuk eshte reference, ne rastin e vargjeve permbajtja ndryshon! */  
echo "Vlera e anetarit te pare te vektorit arr[0]=" . $arr[0]. "<br>";  
echo "Vlera e anetarit te pare te vektorit arr2[0]=" . $arr2[0]. "<br>";?>
```

Përcjellja përmes Referencës

Realizohet duke ndikuar në variable lokale në një funksion të referencojë në të njëjtën përmbajtje me një variabël jashtë funksionit.

Detyra 5: Jepni një shembull të përcjelljes së vlerës me referencë.

```
<?php  
function foo(&$var){  
    $var++;  
}  
$a=5;  
foo($a);  
echo "Vlera e variables a:". $a;  
?>
```

Detyra 6: Jepni një shembull të funksioneve të cilat pranojnë referencë:

```
<?php
    function foo(&$var){
        $var++;
        echo "Vlera e variables var: ".$var;
    }
    function &bar(){
        $a = 5;
        return $a;
    }
    foo(bar());
?>
```

Detyra 7: Shembuj të përcjelljes invalide të referencës në funksione të cilat pranojnë referencë:

```
<?php
    function foo(&$var){
        $var++;
    }
    function bar()    // Vereni mungesën e &
    {
        $a = 5;
        return $a;
    }
    foo(bar());      // Paraqet fatal error qysh prej PHP 5.0.5
    foo($a = 5);     // Shprehje, jo variable
    foo(5);          // Paraqet fatal error
?>
```

Kthimi nëpërmes Referencës

Detyra 8:

```
<?php
    $var = 1;
    $num = NULL;

    function &blah(){
        $var =& $GLOBALS["var"]; # i qasemi variables globale $var;
        $var++;
        return $var;
    }

    $num = &blah();
    echo $num."<br>"; # 2

    blah();
    echo $num; # 3
?>
```

Largimi i PHP Referencës

Detyra 9: Me largimin e referencës, ne këpusim lidhjen ndërmjet emrit të variablës dhe përmbajtjes së saj. Kjo nuk nënkupton që përmbajtja do të largohet.

```
<?php
    $a = 1;
    $b =& $a;
    unset($a);      //kjo komande nuk do te beje unset b vetem a
    echo $b;
    echo $a;        //do të ktheje error, pasi emri i variables a nuk ekziston më
?>
```

Shembull 1: Përshkrim detal mekanizmit të referencave në PHP.

```
<?php
/* Imagjoni sikur kjo eshte nje "memory map"
```

pointer	value	variable	
1	NULL	---	
2	NULL	---	
3	NULL	---	
4	NULL	---	
5	NULL	---	

```
Krijojme disa variabla */
$a=10;
$b=20;
$c=array ('one'=>array (1, 2, 3));
/* Shikojme ne memorie
```

pointer	value	variable's	
1	10	\$a	
2	20	\$b	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2]	

```
do */
$a=&$c['one'][2];
```

/* Shikojme ne memorie

pointer	value	variable's	
1	NULL	---	//...
2	20	\$b	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2] , \$a	// \$a eshte tani ketu

do */

\$b=&\$a; // or \$b=&\$c['one'][2]; rezultate eshte I njejte, sepse te dyja "\$c['one'][2]" and "\$a" jane ne te njejtin pointer

/* Shikojme ne memorie

pointer	value	variable's	
1	NULL	---	
2	NULL	---	//...
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2] , \$a , \$b	// \$b eshte tani ketu

next do */

unset(\$c['one'][2]);

/* Shikojme ne memorie

pointer	value	variable's	
1	NULL	---	
2	NULL	---	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$a , \$b	// ...

next do*/

\$c['one'][2]=500; //tani si array

/* Shikojme ne memorie

pointer	value	variable's	
1	500	\$c['one'][2]	//..
2	NULL	---	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$a , \$b	//ky pointer eshte ne shfrytezim

Tentojme te kthjeme \$c['one'][2] tek pointeri I vjeter dhe keshtu te largojme referencat \$a,\$b.*/

\$c['one'][2]=&\$a;

unset(\$a);

unset(\$b);

/* Shikojme ne memorie

pointer	value	variable's	
1	NULL	---	
2	NULL	---	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2]	// \$c['one'][2] eshte kthyer, \$a,\$b eshte shkaterruar

----- ?>

SQL Injection

Rëndom, qëllimi i SQL injection sulmi, është ekspozimi apo dëmtimi i të dhënave të një ueb-i. Nëpërmes të SQL Injection një sulmues mund të përfitojë nga qasja e paautorizuar në bazë të shënimeve dhe të krijojë, ndryshojë apo fshijë të dhënat. Tani për tani shumica e SQL bazat e të dhënave si MySQL, Oracle, MSSQL Server, PostgreSQL janë të rrezikuara nga SQL Injektion sulmet.

Më poshtë do japim disa prej SQL Injection sulmeve të mundshme në bazë të shënimeve.

Detyra 1: Demonstrimi i SQL Injection permes nje log-in Forme.

```
<?php
$dbhost = 'localhost:3306';
$dbuser = 'root';
$dbpass = '';
$db_name='ushtimet';
$conn = mysql_connect($dbhost, $dbuser, $dbpass)
    or die("cannot connect");

mysql_select_db($db_name)
    or die("cannot select DB");

$uid = ($_POST["user"]);
$pid = ($_POST["pass"]);

$sql = "select * from studentet where ID = '$uid' and password =
'$pid'";

$result = mysql_query( $sql, $conn );
if(mysql_num_rows($result)>0)
{
    while ($row=mysql_fetch_row($result)){
        echo "<h4>". "-- Informatat e studentit me ID:
".$row[0]. "</h4>";
        echo "<p>". "Emri: ".$row[1]. "</p>";
        echo "<p>". "Mbiemri: ".$row[2]. "</p>";
        echo "<p>". "Password: ".$row[3]. "</p>";
        echo "<p>". "Data e regjistrimit: ".$row[4]. "</p>";
        echo "-----";
    }
} else echo "Invalid user id or password";
```

```
        echo "<br>";
        echo $sql;
        echo "<br>";
        echo "-----";
    ?>
<!doctype html>
<html>
<head><title>SQL Injection</title> </head>
<body>
    <form action="<?php $_PHP_SELF ?>" method="POST">
        <label>User ID:</label>
        <input type="text" id="uid" name="user"
placeholder="Student ID" required>
        <label>Password:</label>
        <input type="password" id="passid" name="pass" required>
        <input type="submit" value="Submit" />
    </form>
</body>
</html>
```

Struktura e tabelës: Studentet(Id, emri, mbiemri, password, data_regjistrimit)

Të dhënat:

sid	emri	mbiemri	password	data_regjistrimit
1	Filan	Fisteku	Ffisteku	2014-04-24
2	Barak	Obama	bobama	2014-04-24
3	Fransua	Hollond	fhollond	2014-04-24

Shembull: Leximi i të gjithë shfrytëzuesve nga tabela Studentet nëpërmes SQL Injection në fushat e Login formës së krijuar më lartë.

Duke i shtuar një ' OR '1'='1 në fushën e passwordit atëhere query nga detyra jonë do marrë këtë formë:

"select * from studentet where ID = \$sid and password = " OR '1'='1"

Ky query do të kthejë TRUE për të gjitha rreshtat, për shkak se kushti ose 1=1 do të ndikojë që WHERE të kthejë gjithmonë TRUE.