**MODUL 8 PEMROGRAMAN BERBASIS WEB**

**KONSEP DASAR PHP**

1. **php08A.php** (<https://github.com/afrzl/Pemrograman-Berbasis-Web/blob/8df816d836937af792ff0cab441275f4aaff7939/P8/php08A.php>)

<!DOCTYPE html>

<html lang='en-GB'>

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta name="author" content="Muh. Nur Afrizal">

    <title>Hello World</title>

</head>

<body>

    <h1>Our first PHP script</h1>

    <?php

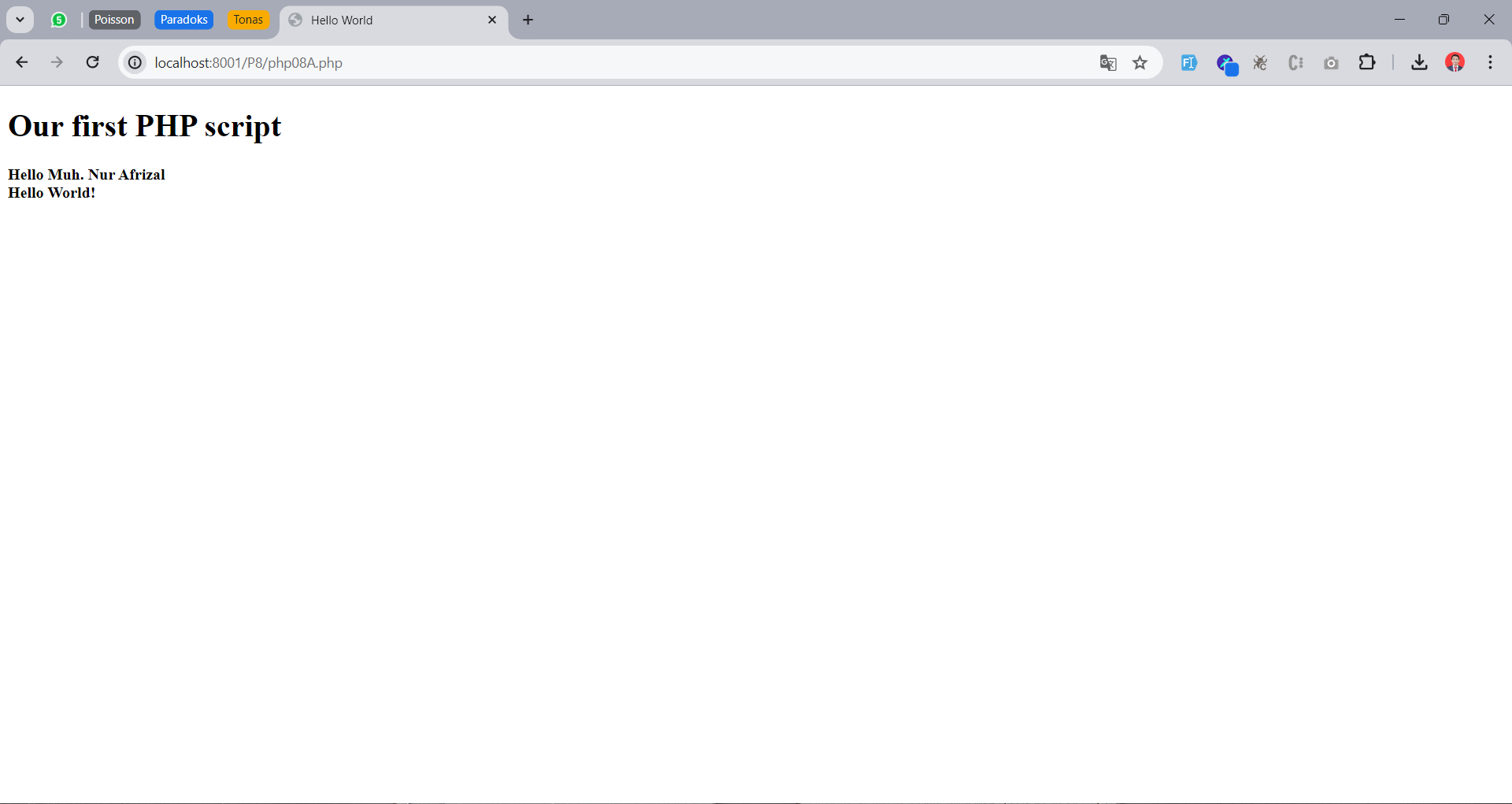
    $user = 'Muh. Nur Afrizal';

    print "<p><b>Hello $user<br>\nHello World!</b></p>\n"; ?>

</body>

</html>

Output:



1. **php08B.php** (<https://github.com/afrzl/Pemrograman-Berbasis-Web/blob/8df816d836937af792ff0cab441275f4aaff7939/P8/php08B.php>)

<!DOCTYPE html>

<html lang='en-GB'>

<head>

    <title>PHP 08B</title>

</head>

<body>

    <h1>Hello World</h1>

    <?php

    error\_reporting(E\_ALL);

    ini\_set('display\_errors', 1);

    ini\_set('display\_startup\_errors', 1);

    echo "<h2>Types and Type Casting</h2>\n";

    $mode = rand(1, 4);

    if ($mode == 1) {

        $randvar = rand();

    } elseif ($mode == 2) {

        $randvar = (string) rand();

    } elseif ($mode == 3) {

        $randvar = rand() / (rand() + 1);

    } else {

        $randvar = (bool) $mode;

    }

    echo "Random scalar: $randvar<br>\n";

*//jawaban*

    if (is\_int($randvar)) {

        echo "This is an integer";

    } elseif (is\_float($randvar)) {

        echo "This is a floating-point number";

    } elseif (is\_string($randvar)) {

        echo "This is a string";

    } else {

        echo "I don't know what this is";

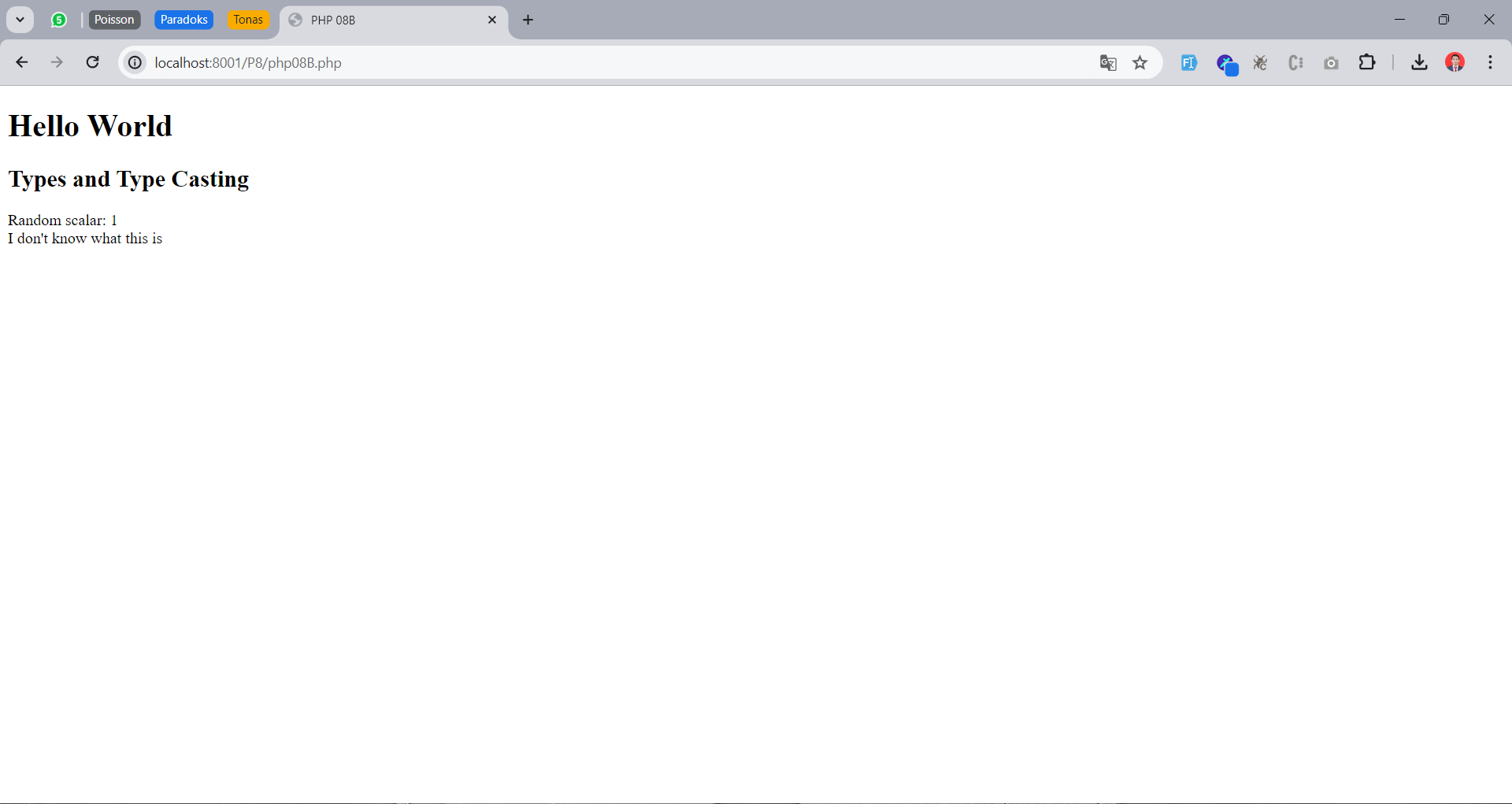
    }

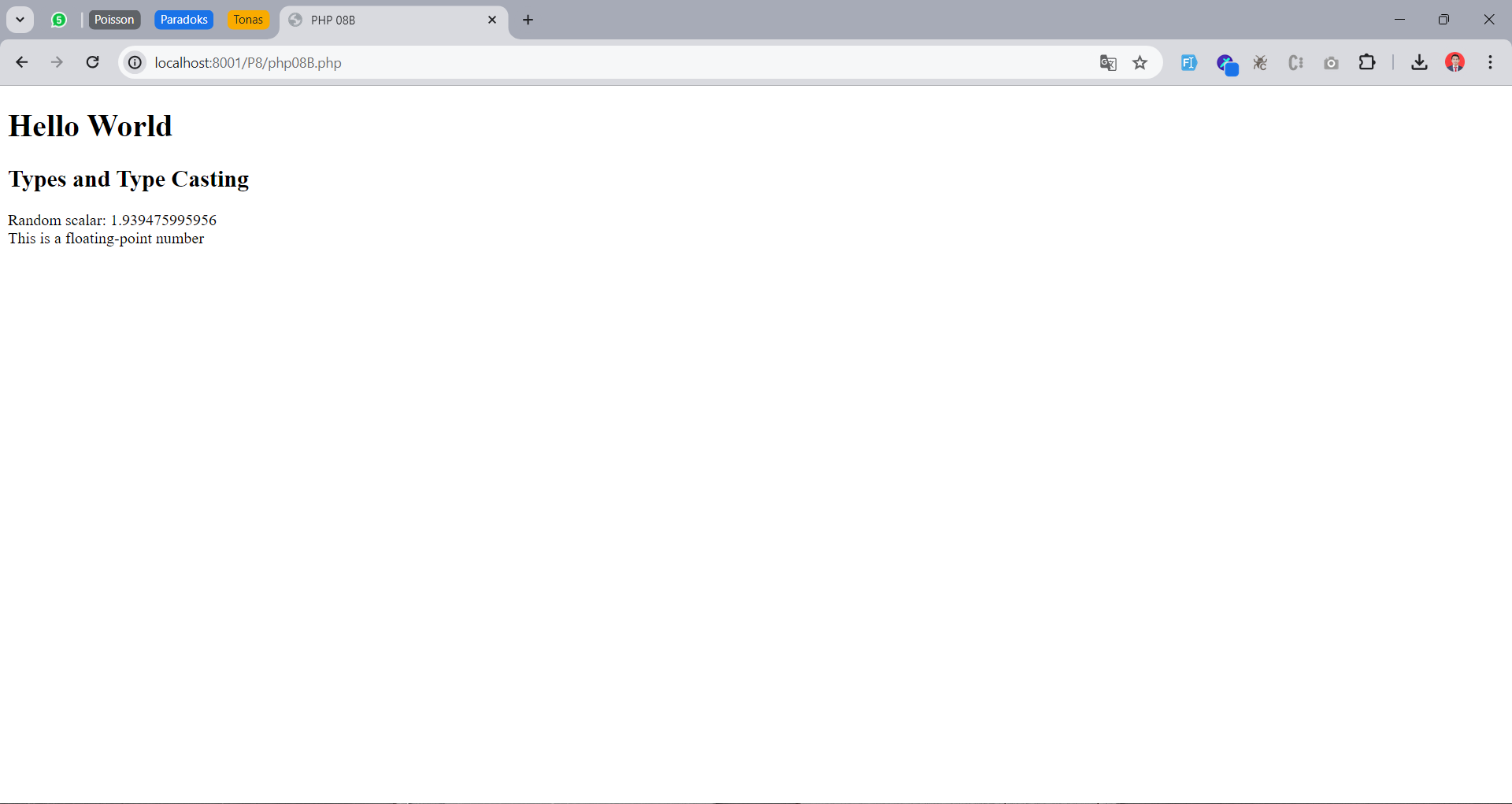
    ?>

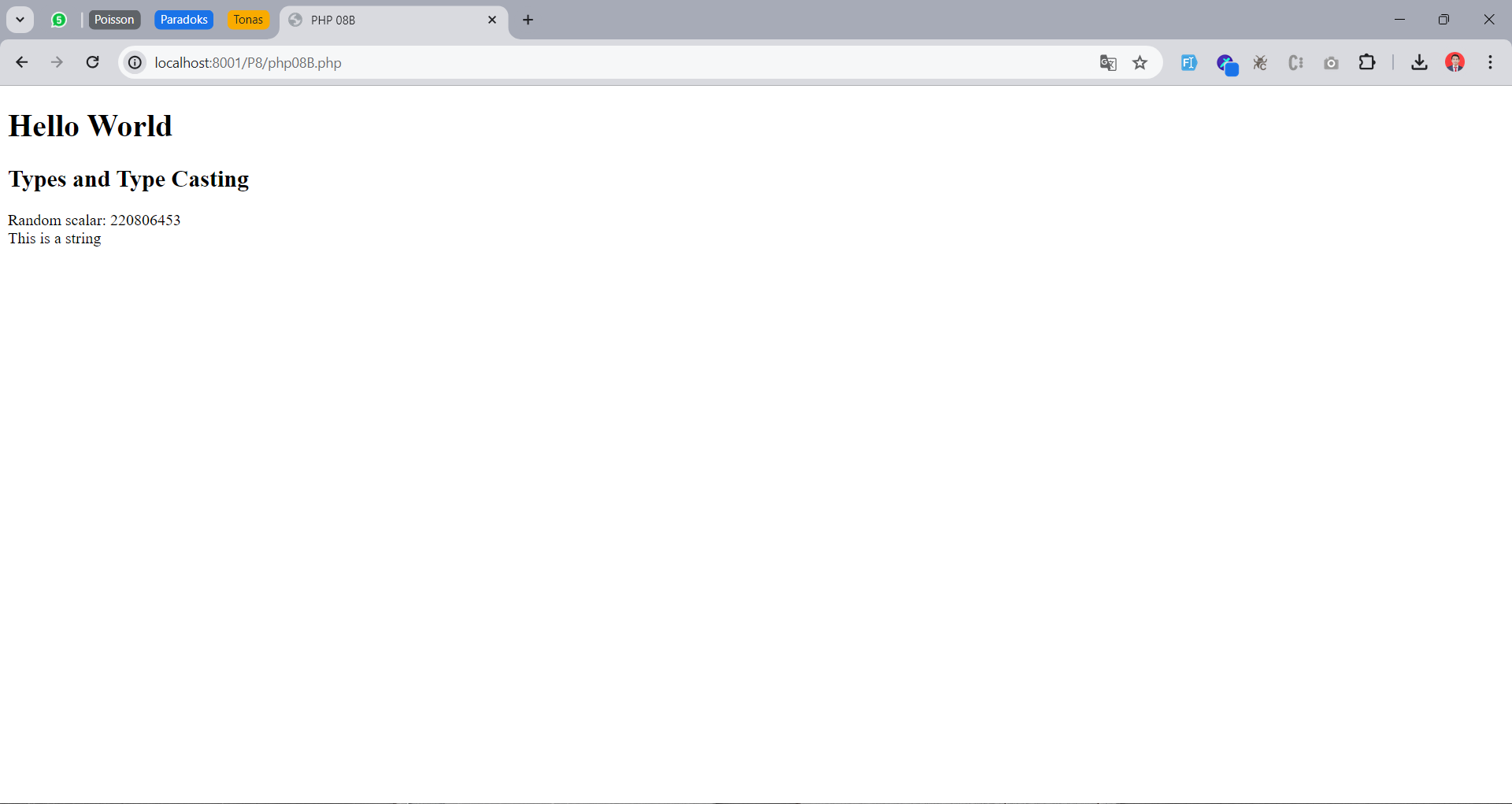
</body>

</html>

Output:







1. **php08C.php** (<https://github.com/afrzl/Pemrograman-Berbasis-Web/blob/8df816d836937af792ff0cab441275f4aaff7939/P8/php08C.php>)

<!DOCTYPE html>

<html>

<head>

    <title>PHP 08C</title>

</head>

<body>

    <h1>Array Operators and Regular Expressions</h1>

    <?php

    echo "<h2>Exercise 2A</h2>\n";

    $planets = ['earth'];

    array\_unshift($planets, 'mercury', 'venus');

    array\_push($planets, 'mars', 'jupiter', 'saturn');

    echo "(1) \$planets = [", join(', ', $planets), "]<br>\n";

    $last = array\_pop($planets);

    echo "(2) \$planets = [", join(', ', $planets), "]<br>\n";

    $first = array\_shift($planets);

    echo "(3) \$planets = [", join(', ', $planets), "]<br>\n";

    echo "(4) \$first = $first, \$last = $last<br>\n";

    echo "<h2>Exercise 2B</h2>\n";

    $spheres = $planets;

    echo "(5) \$spheres = [", join(', ', $spheres), "]<br>\n";

    $planets[1] = 'midgard';

    echo "(6) \$planets = [", join(', ', $planets), "]<br>\n";

    echo "(6) \$spheres = [", join(', ', $spheres), "]<br>\n";

    $spheres = &$planets;

    echo "(7) \$spheres = [", join(', ', $spheres), "]<br>\n";

    $planets[0] = 'friga';

    echo "(8) \$planets = [", join(', ', $planets), "]<br>\n";

    echo "(8) \$spheres = [", join(', ', $spheres), "]<br>\n";

    $arr = $planets;

    foreach ($arr as $key => $value) {

        echo 'Removed: ' . array\_shift($arr) . ' Remaining: ' . '[', join(', ', $arr), "]<br>\n";

    }

    echo "<h2>Exercise 2C</h2>\n";

    $names = ['Dave Shield', 'Mr Andy Roxburgh', 'Dr George Christodoulou', 'Dr Ullrich Hustadt', 'Dr David Jackson'];

    foreach ($names as $name) {

        echo "(1) Name: $name<br>\n";

    }

*// Your own code here*

    $names = str\_replace(['Mr ', 'Dr '], '', $names);

    foreach ($names as $key => $name) {

        $name = explode(" ", $name);

        $names[$key] = strtoupper($name[1]) . ', ' . $name[0];

    }

    foreach ($names as $name)

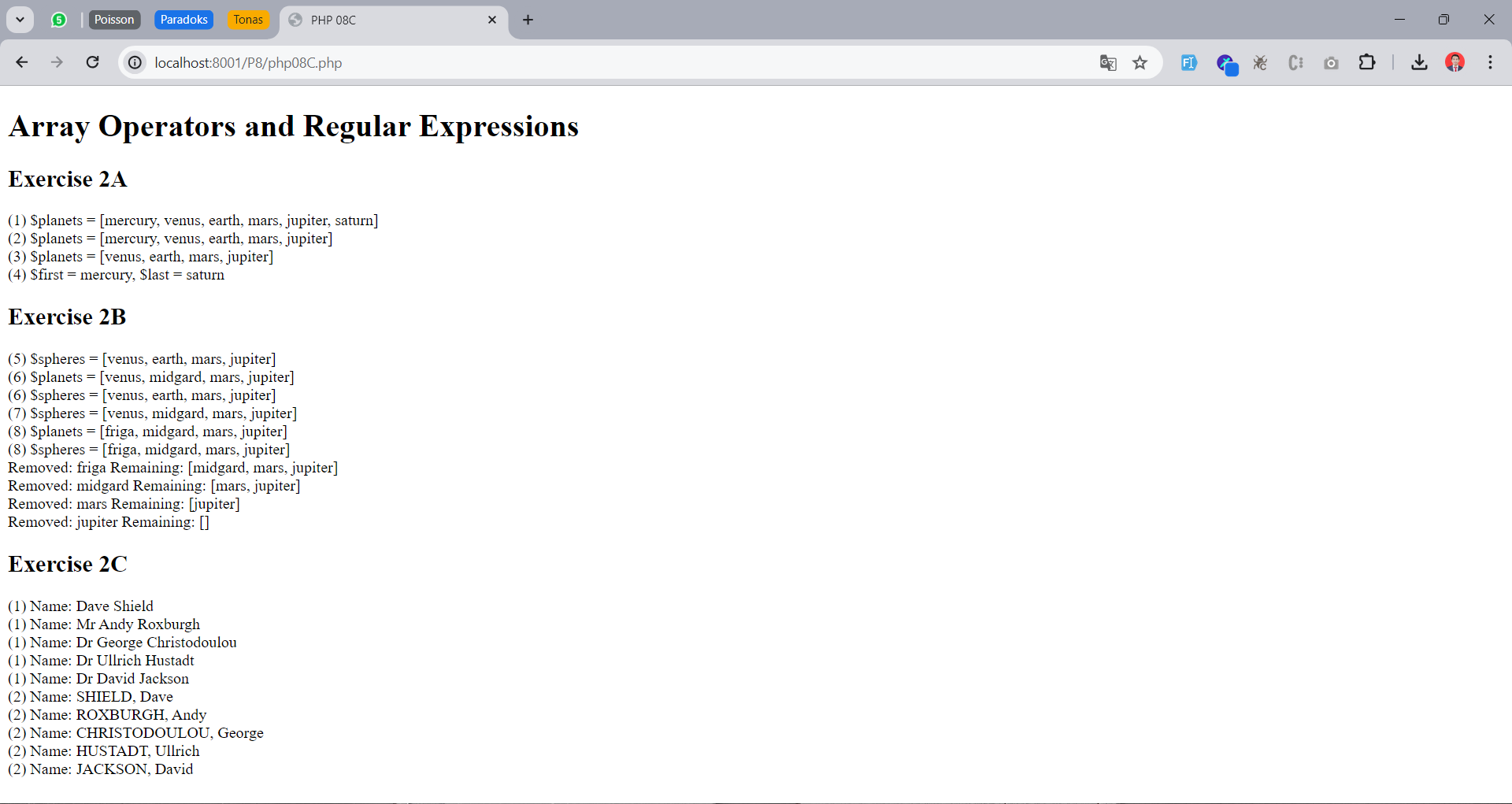
        echo "(2) Name: $name<br>\n"

    ?>

</body>

</html>

Output:



1. **php08D.php** (<https://github.com/afrzl/Pemrograman-Berbasis-Web/blob/8df816d836937af792ff0cab441275f4aaff7939/P8/php08D.php>)

<!DOCTYPE html>

<html>

<head>

    <title>PHP 08D</title>

    <style>

        table, th, td {

            border: 1px solid black;

            border-collapse: collapse;

        }

    </style>

</head>

<body>

    <h1>Associative Arrays</h1>

    <?php

        $dict1 = array('a' => 1, 'b' => 2);

        $dict2 = $dict1;

        $dict1['b'] = 4;

        echo "\$dict1['b'] = ", $dict1['b'],"<br>\n";

        echo "\$dict2['b'] = ", $dict2['b'],"<br>\n";

*//4b*

        foreach ($dict1 as $key => $value) {

            echo "\$dict1['". $key ."'] = ", $value,"<br>\n";

        }

        $text = 'lorem ipsum elit congue auctor inceptos taciti suscipit tortor auctor integer congue hac nullam hac auctor tellus nullam inceptos nullam integer tellus nullam auctor elit lorem ipsum elit';

        $word = str\_word\_count($text, 1);

        $dict3 = array\_count\_values($word);

        krsort($dict3);

        arsort($dict3);

        foreach ($dict3 as $word => $count) {

            echo "$word -> $count <br>\n";

        }

    ?>

    <table>

        <thead>

            <tr>

                <th style="width: 20%">Kata</th>

                <th style="width: 50%">Jumlah Kemunculan</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach($dict3 as $word => $count) { ?>

                <tr>

                    <td><?= $word ?></td>

                    <td style="text-align: right"><?= $count ?></td>

                </tr>

            <?php } ?>

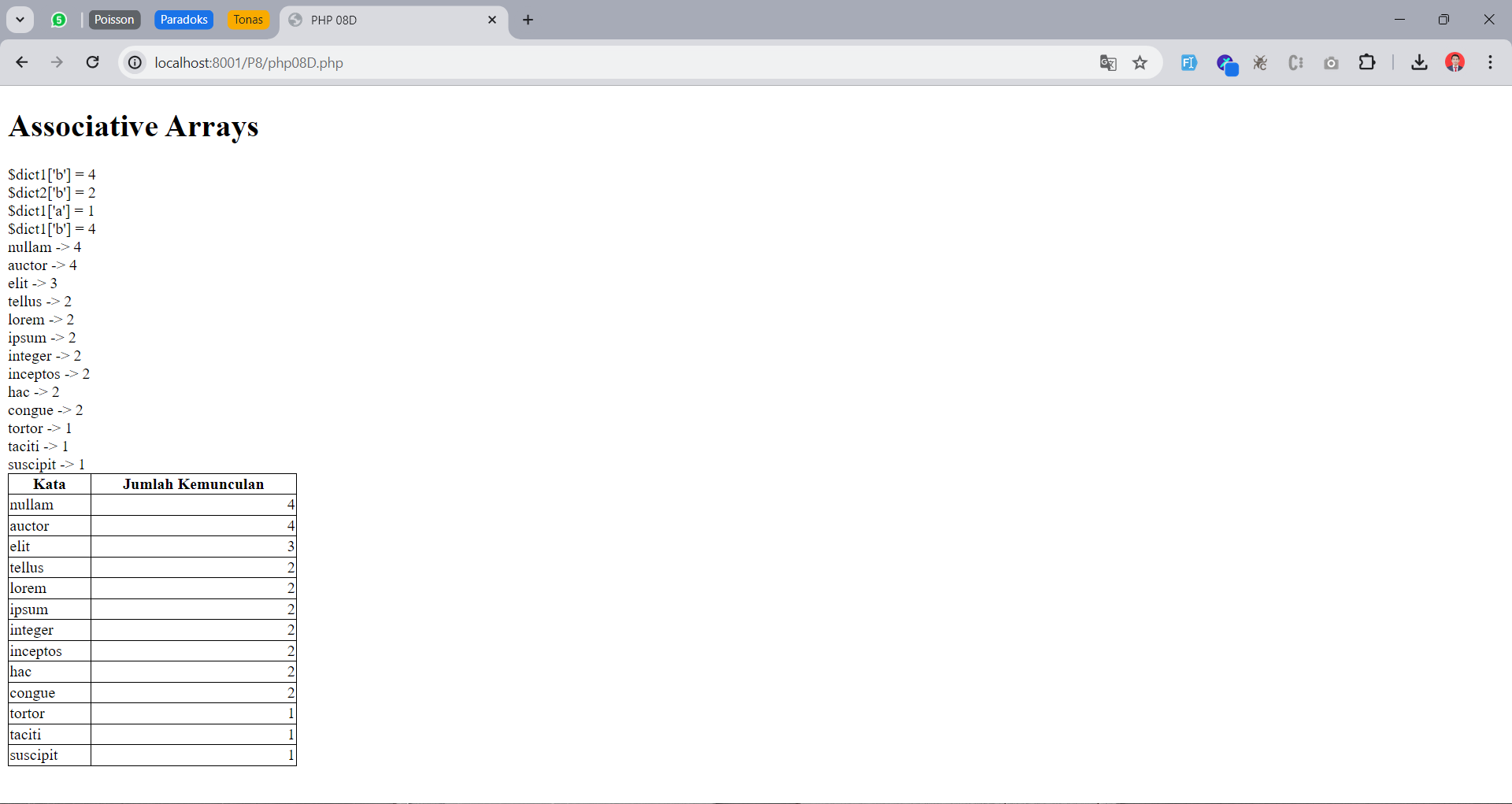
        </tbody>

    </table>

</body>

</html>

Output:



1. **php08E.php** (<https://github.com/afrzl/Pemrograman-Berbasis-Web/blob/8df816d836937af792ff0cab441275f4aaff7939/P8/php08E.php>)

<!DOCTYPE html>

<html lang='en-GB'>

<head>

    <title>PHP 08E</title>

    <META name="description" content="php08E.php">

</head>

<body>

    <h1>Variable-length Argument Lists</h1>

    <?php

        function **reduceOp**(...$op) {

        $last\_index = **array\_key\_last**($op);

        $operator = $op[$last\_index];

        if (!isset($operator["op"])) {

            throw new Exception("TypeError");

        }

        $output = NULL;

        foreach ($op as $key => $value) {

            if ($key != $last\_index) {

                if (!is\_numeric($value)) {

                    throw new Exception("ValueError");

                }

                if ($key == 0) {

                    $output = $value;

                    continue;

                }

                switch ($operator["op"]) {

                    case '+':

                        $output += $value;

                        break;

                    case '-':

                        $output -= $value;

                        break;

                    case '\*':

                        $output \*= $value;

                        break;

                    default:

                        throw new Exception("ValueError");

                        break;

                }

            }

        }

        return $output ? $output : "NULL";

    }

        try {

            echo "1: ", **reduceOp**(2,3), "<br>\n";  *# throws an exception*

        }

        catch (Exception $e) {

            echo "1: Exception ",$e->**getMessage**(),"<br>\n";  *# 'TypeError'*

        }

        try {

            echo "2: ",**reduceOp**(2,3,array('op' => '/')),  *# throws an exception*

                "<br>\n";

        }

        catch (Exception $e) {

            echo "2: Exception ",$e->**getMessage**(),"<br>\n";  *# 'ValueError'*

        }

        echo "3: ",**reduceOp**(array('op'=>'+')),  *# should return NULL*

            "<br>\n";

        echo "4: ",**reduceOp**(2,array('op' => '\*')),  *# should return 2*

            "<br>\n";

        echo "5: ",**reduceOp**(2,3,5,array('op' => '+')),  *# should return 10*

            "<br>\n";

        echo "6: ",**reduceOp**(2,3,5,7,array('op' => '\*')),  *# should return  210*

            "<br>\n";

        echo "7: ",**reduceOp**(2,3,5,7,11,array('op' => '-')),*# should return - 24*

            "<br>\n";

    ?>

</body>

</html>

Output:

