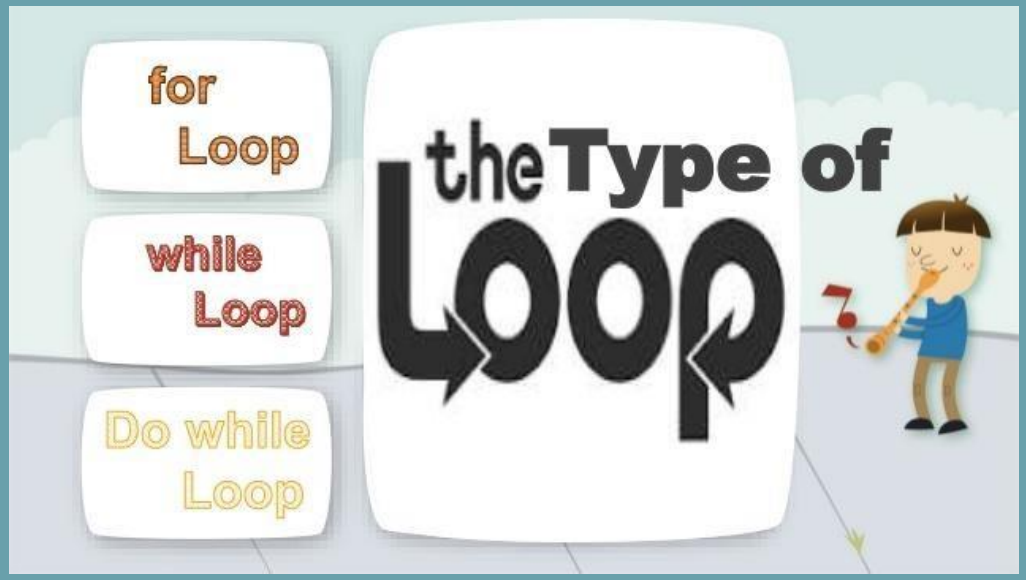
LECTURE 3 – ICT2153

PHP LOOPS



Department of ICT
Faculty of Technology

Objectives

- After the successful completion of this lecture, students should be able to,
  - *Identify and use loops in server side programming*
  - *Identify the use of arrays in web development*
  - *Use arrays in server side programming*

References

- Programming PHP by Kevin Tatroe, Peter MacIntyre and Rasmus Lerdorf

PHP Loops

- Loops execute a block of code a specified number of times, or while a specified condition is true
- PHP have following looping statements ,
  - *while*
  - *do...while*
  - *for*
  - *foreach*

The While Loop

- The while loop executes a block of code while a condition is true

- Ex –

```
<?php
        $num=1;
        while($num<=5)
        {
            echo "The number is " . $num . "<br>";
            $num ++;
        }
    ?>
```

Do-while Loop

- Execute the block of code once, it will then check the condition, and repeat the loop while the condition is true

- Ex –

```
<?php
    $num=1;
    do  {
        $num++;
        echo "The number is " . $num . "<br>";
    }
    while ($num<=5);
?>
```

For Loop

- The for loop is used when you know in advance how many times the script should run

- Ex –

```
<?php
    $counter = 0;
        for($start =0; $start < 11; $start++) {
            $counter = $counter + 1;
            print $counter . "<BR>";
        }
?>
```

The foreach loop

```
<?php
$x=array("Cat","Dog","Hen");
foreach ($x as $value)
{
echo $value . "<br>";
}
?>
```

```
<?php
$x=array("Cat","Dog","Hen");
foreach ($x as $key =>$value)
{
echo $key."-".$value . "<br>";
}
?>
```

Exercises

Create a script that displays 1-2-3-4-5-6-7-8-9-10 on one line. There will be no hyphen(-) at starting and ending position

Answer

```
<?php
 for($x=1; $x<=10; $x++) {
    if($x< 10) {
            echo "$x-";
            } else{
                    echo "$x"."\n";
} }
?>
```

Exercises

Create a script using a for loop to add all the integers between 0 and 30 and display the total

Answer

```
<?php
    $sum = 0;
     for($x=1; $x<=30; $x++) {
          $sum +=$x; }
 echo "The sum of the numbers 0 to 30 is $sum"."\n";
 ?>
```

Exercises

Write a program to calculate and print the factorial of a number using a for loop. The factorial of a number is the product of all integers up to and including that number, so the factorial of 4 is 4*3*2*1= 24

Answer

```
<?php
   $n = 6;
   $x = 1;
    for($i=1;$i<=$n-1;$i++) {
                    $x*=($i+1); }
echo "The factorial of $n = $x"."\n";
?>
```

Exercises

Write a program which will give you all of the potential combinations of a two-digit decimal combination, printed in a comma delimited format :

Sample output :
00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,

Answer

```
<?php
    for($a=0; $a< 10; $a++) {
        for($b=0; $b< 10; $b++) {
            echo $a.$b.", "; }
        }
   printf("\n");
?>
```

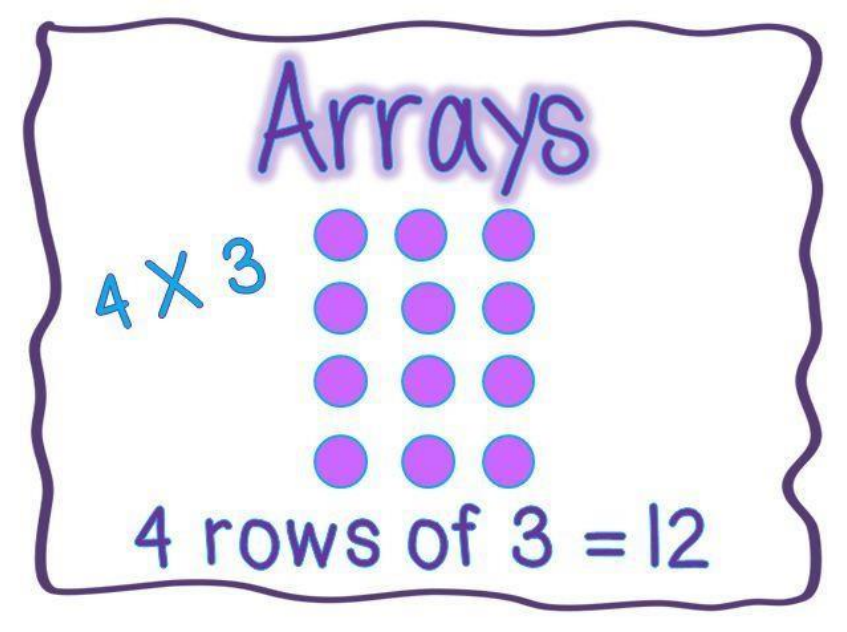
Exercises

Write a PHP program which iterates the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz“

Answer

```
<?php
    for ($i = 1; $i <= 100; $i++) {
    if ( $i%3 == 0 && $i%5 == 0 ) {
        echo $i . " FizzBuzz"."\n" ;
        } else if ( $i%3 == 0 ) {
                echo $i. " Fizz"."\n";
        } else if ( $i%5 == 0 ) {
         echo $i. " Buzz"."\n";
                } else {
                 echo $i."\n"; } }
?>
```

PHP ARRAYS



Department of ICT
Faculty of Technology

Arrays

- An array is a special variable, which can hold more than one value at a time
- It is a collection of data values organized as an ordered collection of key-value pairs
- Arrays store group of related data called '**Elements**'
- An array can hold many values under a single name, and you can access the values by referring to an index number
- Can store heterogeneous data in an array
  - *An array is not limited to one type of data. It can hold strings, integers, Booleans, and so on*

Examples

1.

```
<?php
        $colors = array(“Red", “Green", “Blue");
        echo “Colors are” . $colors[0] . ", " . $colors [1] . " and " . $colors [2] ;
      ?>
```

2.

```
<?php
        $array = array("abc",'a',100,200);
        print "first ".$array[0]." second ".$array[1]." third ".$array[2]." forth
".$array[3];
?>
```

Associated Arrays

```
<?php
    $array = array( "foo" => "bar", "bar" => "foo",  100   => -100,
  -100 => 100);
    print "first ".$array['foo']." second ".$array['bar']." third“
    .$array[100]." forth ".$array[-100];
?>
```

Out put = ????

Multidimensional Array

- Multidimensional array is an array containing one or more arrays.
- In a multidimensional array, The values in an array can themselves be arrays.
- The dimension of an array indicates the number of indices need to select an element.
  - *For a two-dimensional array need two indices to select an element*
  - *For a three-dimensional array need three indices to select an element*

```
$row0 = array(1, 2, 3);
$row1 = array(4, 5, 6);
$row2 = array(7, 8, 9);
$multi = array($row0, $row1, $row2);
```

Arrays of Arrays

- The elements of an array can be many things other than a string or integer
- You can even have objects or other arrays as array elements

```
$products = array(
    'paper' =>    array(
        'copier' => "Copier & Multipurpose",
        'inkjet' => "Inkjet Printer",
        'laser' => "Laser Printer",
        'photo' => "Photographic Paper"),
    'pens' => array(
        'ball' => "Ball Point",
        'hilite' => "Highlighters",
        'marker' => "Markers"),
    'misc' => array(
        'tape'       => "Sticky Tape",
        'glue'       => "Adhesives",
        'clips' => "Paperclips")
);
echo $products["pens"]["marker"];
```

???

Traversing Arrays

- The most common task with arrays is to do something with every element
- Ex:
  - *Sending mail to each element of an array of addresses*
  - *Updating each file in an array of filenames*
- The way of traversing through an array, depends on the data and the task you're performing

Loop Through Indexed Arrays

- You can use a for loop to count through the indices
- The for loop operates on the array itself and processes elements in key order regardless of their internal order

```
<?php
    $color=array("Red","Blue","Green");
    $arrlength=count($color);
    for($x=0;$x<$arrlength;$x++)
    {
        echo $color[$x];
        echo "<br>";
    }
?>
```

Loop Through an Associative Array

- The most common way to loop over elements of an array is to use the **foreach** construct
- Elements are processed by their internal order

```
<?php
    $marks=array("Ruwan"=>35,"Saman"=>37,"Ravi"=>43);
    foreach($marks as $x=>$x_value)
    {
        echo "Student Name=" . $x . ", Marks=" .
    $x_value;
        echo "<br>";
    }
?>
```

Loop Through an Associative Array

- The most common way to loop over elements of an array is to use the **foreach** construct
- Elements are processed by their internal order

```
<?php
    $marks=array("Ruwan"=>35,"Saman"=>37,"Ravi"=>43);
    foreach($marks as $x=>$x_value)
    {
        echo "Student Name=" . $x . ", Marks=" .
    $x_value;
        echo "<br>";
    }
?>
```

```
Student Name=Ruwan, Marks=35
Student Name=Saman, Marks=37
Student Name=Ravi, Marks=43
```

Loop Through a Multidimensional Array

```
<?php
$shop = array( array("rose", 1.25 , 15), array("daisy", 0.75 , 25),
array("orchid", 1.15 , 7) );
for ($row = 0; $row < 3; $row++) {
        echo "<b>The row number $row</b>";
        echo "<ul>";
        for ($col = 0; $col < 3; $col++) {
                echo "<li>".$shop[$row][$col]."</li>";
         }
        echo "</ul>";
}
?>
```

Output

The row number 0

- rose
- 1.25
- 15

The row number 1

- daisy
- 0.75
- 25

The row number 2

- orchid
- 1.15
- 7

Array Functions

Function	Task
count()	Get the length of the array
current()	Returns the current element in an array
next()	Advance the internal array pointer of an array
reset()	Sets the internal pointer of an array to its first element
sort()	Sorts an array

Get the Length of an Array

- The **count()** function is used to return the length (the number of elements) of an array

```
<?php
    $cars = array("Volvo", "BMW", "Toyota");
    echo count($cars);
?>
```

Calculating the Sum of an Array

- The array_sum() function adds up the values in an indexed or associative array
  - ***$sum = array_sum(array);***
  - ***Ex:***

```
$scores = array(98, 76, 56, 80);
$total = array_sum($scores); // $total = 310
```

Inserting an Element Into the End of an Array

- The **array_push()** function inserts one or more elements to the end of an array

```
<?php
    $a=array("red","green");
    array_push($a,"blue","yellow");
    print_r($a);
?>
```

Deleting From an Array

- Deleting an element from an array is just like getting rid of an assigned variable, by calling the **unset()** construct
  - *unset($array_1[2]);*
  - *unset($array_2['yellow']);*
    - *Ex:*

```
<?php
        $anArray = array("X", "Y", "Z");
        unset($anArray[0]);
        print_r($anArray);
?>
```

Deleting From an Array

- Deleting an element from an array is just like getting rid of an assigned variable, by calling the **unset()** construct
  - *unset($array_1[2]);*
  - *unset($array_2['yellow']);*
    - *Ex:*

```
<?php
        $anArray = array("X", "Y", "Z");
        unset($anArray[0]);
        print_r($anArray);
?>
```

```
Array ( [1] => Y [2] => Z )
```

Inspecting Arrays

Function	*Behavior*
is_array()	Takes a single argument of any type and returns a true value if the argument is an array, false otherwise
in_array()	Takes 2 arguments, the **element** you are looking for and the **array** it may be in. If the element is contained as a value in the array, it returns true, otherwise false
IsSet($array[$key])	Takes an array[key] form and returns true if the key portion is a valid key for the array

Questions.....