HTML/PHP - Lecture 9

COEN 10

Sets of data

★Computers deal with large amounts of data

★Sets of data may need to be grouped together

★Data can have

- ◎Same type → Arrays
- ◎Different types → Structures

Sets of data

★Same type → Arrays

- ◎May be very large
- ◎Functionality
  - ✧To access specific elements
  - ✧To search for elements
  - ✧To traverse elements

Arrays

★A variable is a storage area holding a value.

- ◎Problem
  - ✧A variable will hold only one value.
- ◎Solution
  - ✧A special variable, which can hold multiple values, or elements, in one single variable.

Arrays

★An array stores multiple values in one single variable.

- ◎holds all your variable values under a single name
  - ✧Array name → to access the array
  - x[3] ✧Index → to access specific elements

Arrays

★In PHP, there are three kind of arrays:

- ◎Numeric array
  - ✧An array with a numeric index
- ◎Associative array
  - ✧An array where each ID key is associated with a value
- ◎Multidimensional array
  - ✧An array containing one or more arrays

Numeric Arrays

★ A numeric array uses numeric indices to access elements.

★ There are two methods to create a numeric array.

1. Automatically (index starts at 0):

```
$vacation=array (“Rio”, “Hawaii", “New York”, “Paris”);
```

2. Manually:

```
$vacation[0]=“Rio”;
$vacation[1]=“Hawaii”;
$vacation[2]=“New York”;
$vacation[3]=“Paris”;
```

Note: Array $vacation has 4 elements, the first one is in position zero and the last one is in position 3.

Numeric Arrays

★Example

```
<?php
  $vacation[0]=“Rio”;
  $vacation[1]=“Hawaii”;
  $vacation[2]=“New York”;
  $vacation[3]=“Paris”;
  echo $vacation[0] . " and " . $vacation[1] .
   “have perfect beaches.”;
?>
```

Array - Functions

★ count

```
count ($array);
```

returns the number of elements in the array

★ push

```
push ($array, element, element, …);
```

adds elements at the end of the array

★ pop

```
$variable = pop ($array);
```

extract the last element from the array and assign to $variable

Array - Functions

★ More at

http://www.w3schools.com/php/php_ref_array.asp

Numeric Arrays - Examples

★ Creating an array of numbers

1. Automatically:

```
$x = array (216, 45, 56, 10000);
```

2. Manually:

```
$x[0] = 216;
$x[1] = 45;
$x[2] = 56;
$x[3] = 10000;
```

Numeric Array - Examples

★ Elements can be manipulated independently

★ Examples

216

◎ Assign the 1st element to variable $y

```
$y = $x[0];
```

◎ Assign value 10 to the last element

```
$size = count($x);
$x[$size - 1] = 10;
```

56

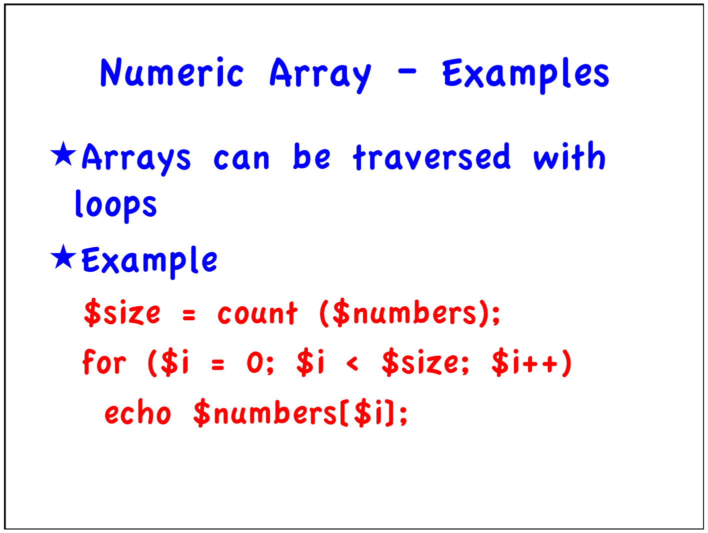
◎ Output the 3rd element

```
echo $x[2];
```

216+34
=261

◎ Add the two first elements and assign the result to variable $z

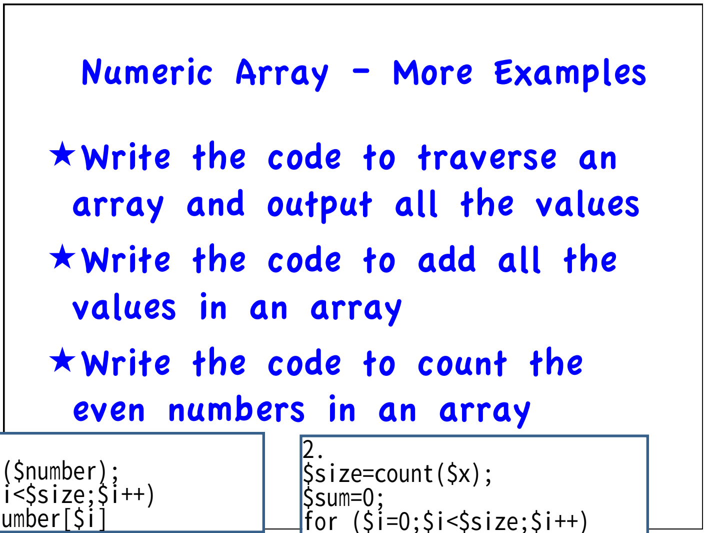
```
$z = $x[0] + $x[1];
```



Numeric Array - Examples

- ★Arrays can be traversed with loops
- ★Example

```
$size = count ($numbers);
for ($i = 0; $i < $size; $i++)
 echo $numbers[$i];
```



Numeric Array - More Examples

- ★Write the code to traverse an array and output all the values
- ★Write the code to add all the values in an array
- ★Write the code to count the even numbers in an array

```
1.
$size=count($number);
for ($i=0;$i<$size;$i++)
    echo $number[$i]
```

```
2.
$size=count($x);
$sum=0;
for ($i=0;$i<$size;$i++)
    $sum+=$x[$i]
```

```
3.
Method1:
$size=count($numbers);
for ($i=0;$i<$size;$i++)
    $size=count($i+=2);

Method2:
$size=count($x);
$count=0;
for($i=0;$i<$size;$i++)
    if ($x[$i]%2==0)
        $count++;
```