APPDET

APPLICATION DEVELOPMENT &
EMERGING TECHNOLOGY

Prof. ROEL C. TRABALLO
roel.traballo@umak.edu.ph

INTRODUCTION TO WEB DEVELOPMENT

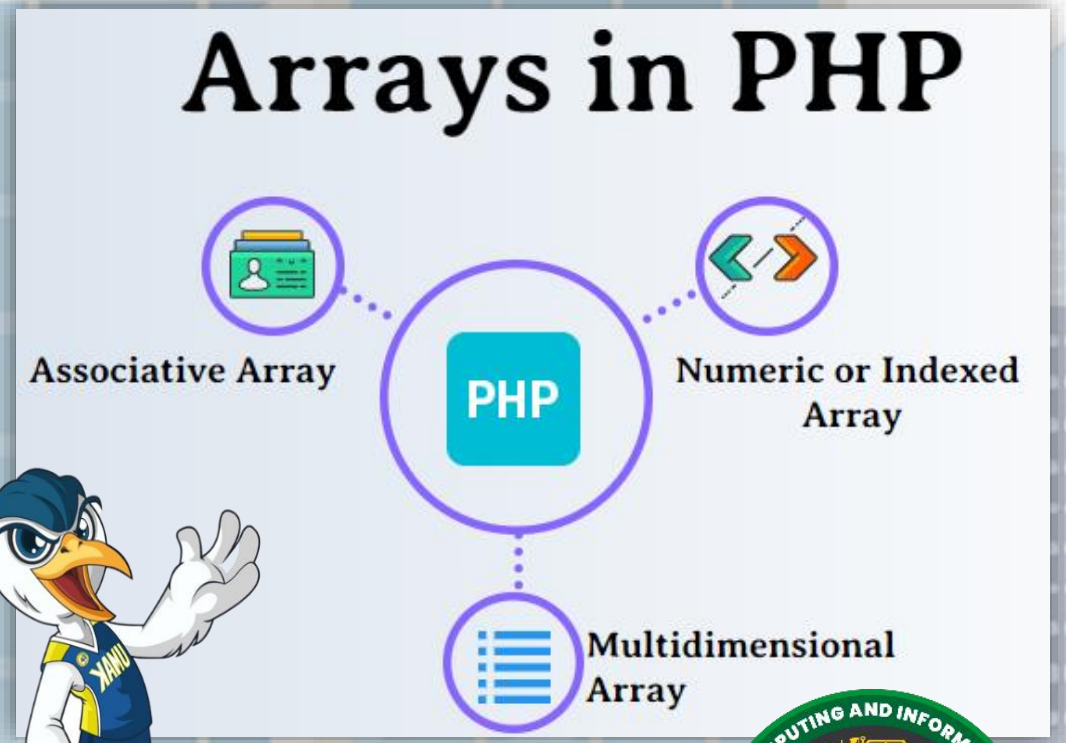
- PHP Arrays and ForEach
- PHP String Functions

WEEK-5

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS

- By definition, an **array** is a list of elements. So, for example, you may have an array that contains a list of products.
- PHP provides you with two types of arrays: indexed and associative.
- The keys of the indexed array are integers that start at 0. Typically, you use indexed arrays when you want to access the elements by their positions.
- The keys of an associative array are strings.
- And you use associative arrays when you want to access elements by string keys.



PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Creating Arrays

- In PHP, you can use the array() construct or [] syntax to define an array.
- The [] syntax is shorter and more convenient.

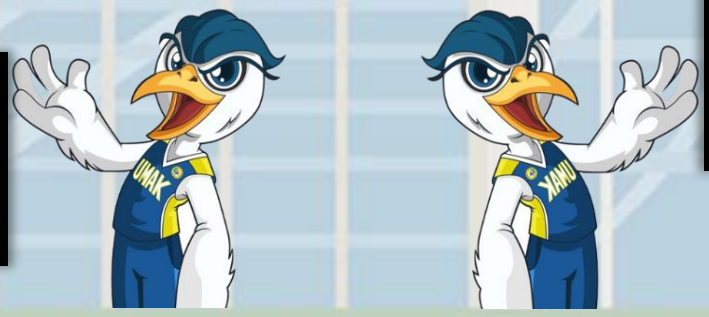
1) Creating an array using **array()** construct

To define an array, you use the array() construct.

✓ The following example creates an empty array:

```
<?php

$empty_array = array();
```



- To create an array with some initial elements, you place a comma-separated list of elements within parentheses of the array() construct.
- For example, the following defines an array that has three numbers:

```
<?php

$scores = array(1, 2, 3);
```

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Creating Arrays

- In PHP, you can use the array() construct or [] syntax to define an array.
- The [] syntax is shorter and more convenient.

2) Creating an array using the [] syntax

PHP provides a more convenient way to define arrays with the shorter syntax [], known as JSON notation. The following example uses [] syntax to create a new empty array:

```
<?php

$empty_array = [];
```

- The following example uses the [] syntax to create a new array that consists of three numbers:

```
<?php

$scores = [1, 2, 3];
```

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Displaying arrays

- To show the contents of an array, you use the **var_dump()** function.

Example:

```
<?php

$scores = [1, 2, 3];
var_dump($scores);
```

Output:

```
array(3) {
  [0]=> int(1)
  [1]=> int(2)
  [2]=> int(3)
}
```

- Or you can use the **print_r()** function:

Example:

```
<?php

$scores = array(1, 2, 3);
print_r($scores);
```

Output:

```
Array
(
    [0] => 1
    [1] => 2
    [2] => 3
)
```

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Displaying arrays

- To make the output more readable, you can wrap the output of the print_r() function inside a <pre> tag.

Example:

```
<?php

$scores = [1, 2, 3];

echo '<pre>';
print_r($scores);
echo '</pre>';
```

Output:

```
Array
(
    [0] => 1
    [1] => 2
    [2] => 3
)
```

- It's more convenient to define a function that prints out an array like this:

Example:

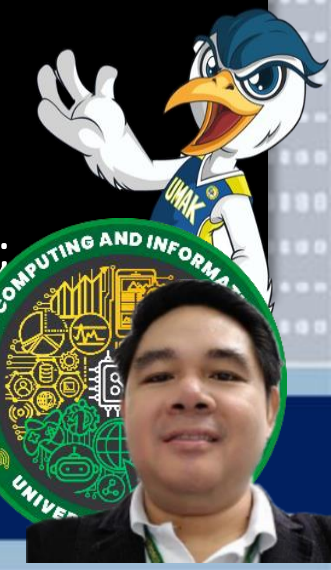
```
<?php

function print_array($data) {
    echo '<pre>';
    print_r($data);
    echo '</pre>';
}

$scores = [1, 2, 3];
print_array($scores);
```

Output:

```
Array
(
    [0] => 1
    [1] => 2
    [2] => 3
)
```



PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Accessing array elements

- To access an element in an array, you specify the index of the element within the square brackets:
- **Syntax:**

```
$array_name[index]
```

- Note that the index of the first element of an array begins with zero, not one.

Example:

```
<?php

$scores = [1, 2, 3];
echo $scores[0];
```

Output:

```
1
```

- Adding an element to the array
- **Syntax:**

```
$array_name[] = new_element;
```

Example:

```
<?php

$scores = [1, 2, 3];
$scores[] = 4;
```

- In this example, we defined an array that consists of three numbers initially.
- Then, we added the number 4 to the array.
- But doing this, you have to calculate the new index manually. It is not practical.
- Also, if the index is already is used, the value will be overwritten.

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Changing array elements

- The following statement changes the element located at the index to the $new_element:
- **Syntax:**

```
$array_name[index] = $new_element;
```

- Note that the index of the first element of an array begins with zero, not one.

Example:

```
<?php

$scores = [1, 2, 3];
echo $scores[0];
```

Output:

```
1
```

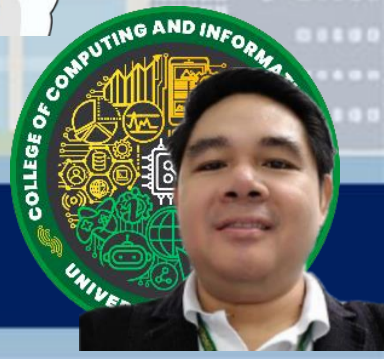
Removing array elements

- To remove an element from an array, you use the unset() function.
- The following removes the second element of the $scores array:

Example:

```
<?php

$scores = [1, 2, 3];
unset($scores[1]);
```



PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Getting the size of an array

- To get the number of elements in an array, you use the count() function. For example:

Example:

```
<?php

$scores = [1, 2, 3, 4, 5];

echo count($scores);
```

Output:

```
5
```

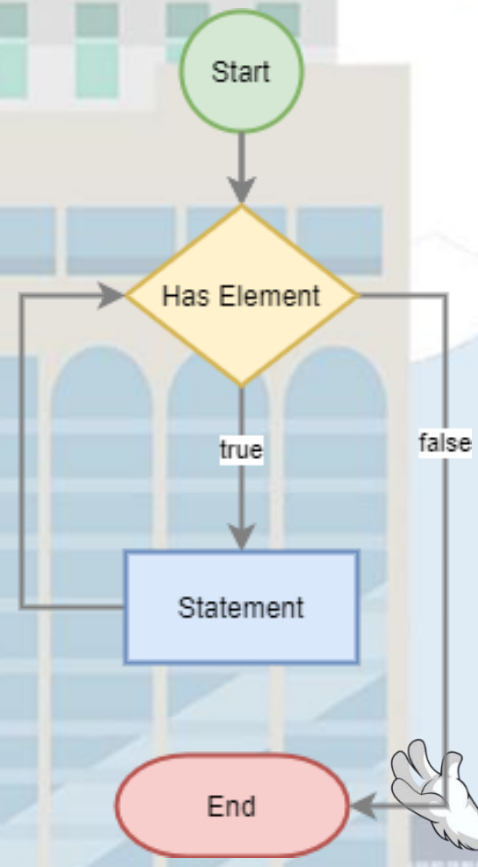
SUMMARY

- Use the array() construct or [] syntax to create a new array.
- For the indexed array, the first index begins with zero.
- To access an array element, use an index in the square bracket $array_name [index].
- Use the count() function to get the number of elements in an array.

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: foreach loop

- PHP provides you with the **foreach** statement that allows you to iterate over elements of an array, either an indexed array or an associative array.
- The foreach statement iterates over all elements in an array, one at a time. It starts with the first element and ends with the last one.
- Therefore, you don't need to know the number of elements in an array upfront.



- **Syntax:**

```
<?php

foreach ($array_name as $element) {
   // process element here
}
```

PHP FUNDAMENTALS: ARRAYS

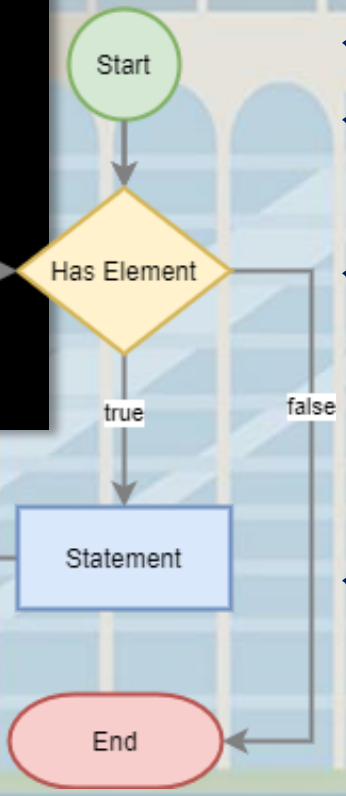
PHP ARRAYS: foreach loop

Example:

```
<?php

$colors = ['red', 'green', 'blue'];

foreach ($colors as $color) {
    echo $color . '<br>';
}
```



Output:

```
red
green
blue
```

- When PHP encounters the **foreach** statement, it accesses the first element and assigns:
  - ✓ The key of the element to the $key variable.
  - ✓ The value of the element to the $**value** variable.
  - ✓ In each iteration, PHP assigns the key and value of the next element to the variables ($key and $value) that follows the as keyword.
  - ✓ If the last element is reached, PHP ends the loop.

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: foreach loop

Example:

```
<?php

$capitals = [
            'Japan' => 'Tokyo',
            'France' => 'Paris',
            'Germany' => 'Berlin',
            'United Kingdom' => 'London',
            'United States' => 'Washington D.C.'];

foreach ($capitals as $country => $capital) {
      echo "The capital city of {$country} is $capital" .'<br>';
}
```

Output:

```
The capital city of Japan is Tokyo
The capital city of France is Paris
The capital city of Germany is Berlin
The capital city of United Kingdom is London
The capital city of United States is Washington D.C.
```

Summary:

- Use the **foreach($array_name as $element)** to iterate over elements of an indexed array.
- Use the **foreach($array_name as $key => $value)** to iterate over elements of an associative array.

PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Associative Arrays

Associative arrays are arrays that allow you to keep track of elements by names rather than by numbers.

- **Creating associative arrays**: To create an associative array, you use the array() construct:

```
<?php

$html = array();
```

- **Adding elements to an associative array:**

Example:

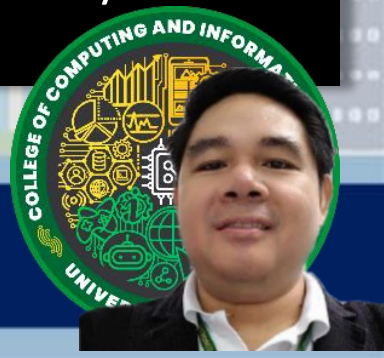
```
<?php

$html['title'] = 'PHP Associative Arrays';
$html['description'] = 'Learn how to use
associative arrays in PHP';

print_r($html);
```

Output:

```
Array
(
    [title] => PHP Associative Arrays
    [description] => Learn how to use associative arrays in PHP
)
```



PHP FUNDAMENTALS: ARRAYS

PHP ARRAYS: Associative Arrays

- **Accessing elements in an associative array:**
- To access an element in an associative array, you use the key.

Example:

```
<?php

$html['title'] = 'PHP Associative Arrays';
$html['description'] = 'Learn how to use associative
arrays in PHP';

echo $html['title'];
```

Output: PHP Associative Arrays

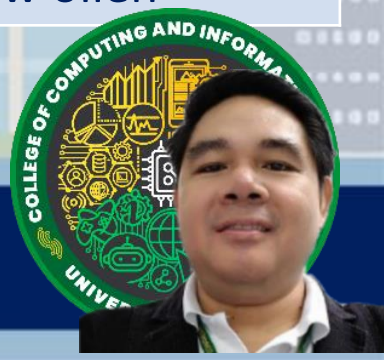
Summary:

- Use an associative array when you want to reference elements by names rather than numbers.

PHP FUNDAMENTALS: STRING FUNCTIONS

Some String Functions

SYNTAX	DESCRIPTION	EXAMPLE	OUTPUT
strlen(string)	returns the length of a string.	strlen(“Hello World”);	11
ucfirst(string)	converts the first character of a string to uppercase	ucfirst(“hello world”);	Hello world
ucwords(string)	converts the first character of each word in a string to uppercase	ucwords(“hello world”);	Hello World
lcfirst(string)	Convert the first character to lowercase	lcfirst(Hello World”)	hello World
strtoupper(string)	converts a string to uppercase	strtoupper (“hello world”);	HELLO WORLD
strtolower(string)	converts a string to lowercase	strtolower(“HELLO WORLD”);	hello world
strrev(string)	reverses a string.	Strrev(“hello world”)	dlrow olleh



APPDET

APPLICATION DEVELOPMENT & EMERGING TECHNOLOGY