PHP ARRAYS

PHP arrays

- Arrays are complex variables that allow us to store more than one value or a group of values under a single variable name.

- Arrays are zero-indexed.

- Why to use arrays?

✓Efficiently manage lists of data.

✓Group related data together.

✓Simplify code by reducing the number of variables needed.

Types of Arrays in PHP

There are three types of arrays that you can create. These are:

- **Indexed array** — An array with a numeric key.
- **Associative array** — An array where each key has its own specific value.
- **Multidimensional array** — An array containing one or more arrays within itself.

Creating and initializing arrays

- Two ways to declare an empty array:
- Declare empty arrays: $emptyArray = array();
- PHP 5.4+ shortcut: $emptyArray = [];

Creating and initializing arrays

- Two ways to declare an indexed array:
- Use array() function : $colors = array("red", "green", "blue");
- Shortcut method in PHP 5.4+: $colors = ["red", "green", "blue"];

Creating and initializing arrays

- Two ways to declare an associative arrays:
- Use array() function with key-value pairs : $person = array("name" => "John", "age" => 30);
- PHP 5.4+ shortcut: $person = ["name" => "John", "age" => 30];

Creating and initializing arrays

- Two ways to declare a multidimensional arrays:

- Nested array() calls.

```php
$matrix = array(
  array(1, 2, 3),
  array(4, 5, 6),
  array(7, 8, 9)
);
```

Accessing Elements

- Use square brackets with index: $colors[0]; // "red“
- Accessing elements in associative arrays: $person['name']; // "John“
- Access elements in multidimensional arrays: $matrix[2][1]; // 8

Modifying Array Elements

- Update values by index in indexed array:
  $colors[0] = "yellow“;
- Update values by key in associative array :
  $person["city"] = "New York";
- Update values by index/key in multidimensional array:
  $person[0]["city"] = "New York";

Removing Elements from Arrays

- Use unset() function to remove the element in indexed arrays:

  unset($colors[2]); // Removes "blue“

- Use unset() function to remove the element in associative arrays:

  unset($person["age"]);

Indexed Arrays

- An indexed or numeric array stores each array element with a numeric index.

```
<?php
$courses = array("PHP", "Laravel", "Node js");
echo "I like " . $courses[0] . ", " . $courses[1] . " and " . $courses[2];
echo "<br>";
echo count($courses);
?>
```

OUTPUT:

I like PHP, Laravel and Node js

3

Loop Through an Indexed Array(for loop)

```
<?php
$courses = array("PHP", "Laravel", "Node js");
$courseslength = count($courses);

for($x = 0; $x <$courseslength; $x++) {
  echo $courses[$x];
  echo "<br>";
}
?>
```

OUTPUT:

PHP

Laravel

Node js

Loop Through an Indexed Array(PHP foreach Loop)

- The foreach loop is used to iterate over arrays.
- It is used to loop through each key/value pair in an array.

```
<?php
$courses = array("PHP", "Laravel", "Node js");

// Loop through colors array
foreach($courses as $course){
    echo $course . "<br>";
}
?>
```

OUTPUT:
PHP
Laravel
Node js

Associative Array

- Associative arrays are arrays that use named keys that you assign to them.
- We can associate name with each array elements in PHP using => symbol.
- The keys assigned to values can be arbitrary and user defined strings.

Associative Array(contd.)

```
<?php
$courses = array("INT220"=>"PHP", "INT221"=>"Laravel", "INT222"=>"Node js");
echo "INT 220 is ".$courses['INT220']."." INT 221 is ".$courses['INT221']."." INT222 is ".$courses['INT222'];
?>
```

OUTPUT:

INT 220 is PHP. INT 221 is Laravel. INT222 is Node js

Associative Array(contd.)

```
<?php
$courses["INT220"] = "PHP";
$courses["INT221"] = "Laravel";
$courses["INT222"] = "Node js";

// Printing array structure
print_r($courses);
?>
```

OUTPUT:

Array ([INT220] => PHP [INT221] => Laravel [INT222] => Node js)

Loop Through an Associative Array(for each loop)

```
<?php
$courses = array("INT220"=>"PHP","INT221"=>"Laravel","INT222"=>"Node js");
foreach($courses as $course => $value) {
  echo "Key=".$course.","."Value=".$value;
  echo "<br>";
}
?>
```

OUTPUT:

Key=INT220, Value=PHP

Key=INT221, Value=Laravel

Key=INT222, Value=Node js

Loop Through an Associative Array(for loop)

```
<?php
$courses = array('INT220'=>'PHP','INT221'=>'Laravel','INT222'=
>'Node js');
$keys = array_keys($courses);
$values = array_values($courses);
for($x=0; $x<count($courses); $x++) {
    echo "Key=".$keys[$x].','."Value=".$values[$x]. "<br>";
}
?>
```

OUTPUT:

Key=INT220,Value=PHP

Key=INT221,Value=Laravel

Key=INT222,Value=Node js

Multidimensional Arrays

- The multidimensional array is an array in which each element can also be an array and each element in the sub-array can be an array or further contain array within itself and so on.

Multidimensional Arrays(contd.)

```
<?php
$result = array(
    array("Manoj",7.8,"pass"),
    array("Aditya",8.5,"pass"),
    array("Anuj",4.9,"fail")
);
echo $result[0][0]. "----CGPA is: " . $result[0][1]." and his status is ".$result[0][2]."<br>";
echo $result[1][0]. "----CGPA is: " . $result[1][1]." and his status is ".$result[1][2]."<br>";
echo $result[2][0]. "----CGPA is: "  . $result[2][1]." and his status is ".$result[2][2];
?>
```

OUTPUT:

Manoj----CGPA is: 7.8 and his status is pass
Aditya----CGPA is: 8.5 and his status is pass
Anuj----CGPA is: 4.9 and his status is fail

Multidimensional Arrays(contd.)

```
<?php
$result = array(
    array(
        "name" => "Manoj",
        "cgpa" => 7.8,
        "status" => "pass"
    ),
    array(
        "name" => "Aditya",
        "cgpa" => 8.5,
        "status" => "pass"
    ),
    array(
        "name" => "Anuj",
        "cgpa" => 4.9,
        "status" => "fail"
    )
);
echo $result[0]["name"]. "----CGPA is: " . $result[0]["cgpa"]." and his status is ".$result[0]["status"]."<br>";
echo $result[1]["name"]. "----CGPA is: " . $result[1]["cgpa"]." and his status is ".$result[1]["status"]."<br>";
echo $result[2]["name"]. "----CGPA is: "   . $result[2]["cgpa"]." and his status is ".$result[2]["status"];
?>
```

OUTPUT:

Manoj----CGPA is: 7.8 and his status is pass

Aditya----CGPA is: 8.5 and his status is pass

Anuj----CGPA is: 4.9 and his status is fail

Loop Through an Multidimensional Array(for loop)

```
<?php
$result = array (
  array("Manoj",7.8,"pass"),
  array("Aditya",8.5,"pass"),
  array("Anuj",4.9,"fail")

);

for ($row = 0; $row < 3; $row++) {
  echo "<h4>Row number $row</h4>";
  for ($col = 0; $col < 3; $col++) {
    echo $result[$row][$col]."<br>";
  }
}
?>
```

OUTPUT:

Row number 0
Manoj
7.8
Pass

Row number 1
Aditya
8.5
Pass

Row number 2
Anuj
4.9
fail

Loop Through an Multidimensional Array(foreach loop)

```
<?php
$result = array (
  array("Manoj",7.8,"pass"),
  array("Aditya",8.5,"pass"),
  array("Anuj",4.9,"fail")
);
for($row = 0; $row < 3; $row++) {
        echo "<h4>Row number $row</h4>";

    foreach ($result[$row] as $resul) {
      echo $resul."<br>";
    }
  }
?>
```

OUTPUT:

Row number 0
Manoj
7.8
Pass

Row number 1
Aditya
8.5
Pass

Row number 2
Anuj
4.9
fail

Loop Through an Multidimensional Array(foreach loop)

```
<?php
$books =
array("C++" => array("name" => "Beginning with C","copies" =>8),
      "PHP" => array("name" => "Basics of PHP","copies" => 10),
      "Laravel" => array("name" => "MVC","copies" => 3)
);

$keys = array_keys($books);
for($i = 0; $i < count($books); $i++) {
    echo "<h1>$keys[$i]</h1>";
    foreach($books[$keys[$i]] as $key => $value) {
        echo $key . " = " . $value . "<br>";
    }
}
?>
```

C++
name = Beginning with C
copies = 8

PHP
name = Basics of PHP
copies = 10

Laravel
name = MVC
copies = 3