**Id code:** 19CM029  **Name:** Dhiraj Jagtap  **Sem:** 5th

**Practical No: 4**

**Aim:** Write a PHP program for creating and manipulating.

a. Indexed array

b. Associative array

c. Multidimensional array

**Theory:**

Arrays in PHP is a type of data structure that allows us to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable for every data. The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key. Suppose we want to store five names and print them accordingly. This can be easily done by the use of five different string variables. But if instead of five, the number rises to a hundred, then it would be really difficult for the user or developer to create so many different variables. Here array comes into play and helps us to store every element within a single variable and also allows easy access using an index or a key. An array is created using an array () function in PHP.

There are basically three types of arrays in PHP:

* **Indexed or Numeric Arrays**: An array with a numeric index where values are stored linearly.
* **Associative Arrays:** An array with a string index where instead of linear storage, each value can be assigned a specific key.
* **Multidimensional Arrays:** An array which contains single or multiple arrays within it and can be accessed via multiple indices.

1. **Indexed or Numeric Arrays:  
   Theory:**

These type of array can be used to store any type of elements, but an index is always a number. By default, the index starts at zero. These arrays can be created in two different ways

**Program:**

<?php

// One way to create an indexed array

$name\_one = array("Rohan", "Ujawal", "Pratham", "Harish", "Pranav");

// Accessing the elements directly

echo "Accessing the 1st array elements directly:<br>";

echo $name\_one[2], "<br>";

echo $name\_one[0], "<br>";

echo $name\_one[4], "<br>";

// Second way to create an indexed array

$name\_two[0] = "ROHAN";

$name\_two[1] = "UJAWAL";

$name\_two[2] = "PRATHAM";

$name\_two[3] = "HARISH";

$name\_two[4] = "PRANAV";

// Accessing the elements directly

echo "Accessing the 2nd array elements directly:<br>";

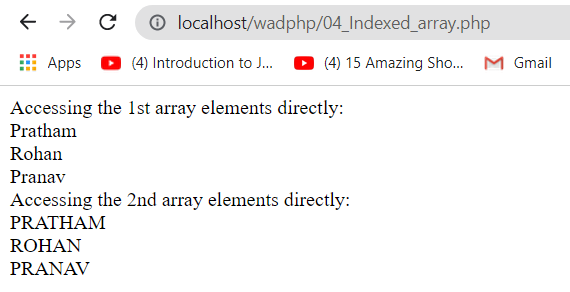
echo $name\_two[2], "<br>";

echo $name\_two[0], "<br>";

echo $name\_two[4], "<br>";

?>

**Output:**



1. **Associative Arrays:**

**Theory:** These types of arrays are similar to the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type.

**Program:**

<?php

// One way to create an associative array

$name\_one = array("Rohan"=>"Mahalle", "Ujjwal"=>"Khairner",

                  "Pratham"=>"Gaur", "Harish"=>"chopade",

                  "Kiran"=>"Mali");

// Second way to create an associative array

$name\_two["Sahil"] = "Asole";

$name\_two["Soham"] = "Borkar";

$name\_two["Pavan"] = "Mule";

$name\_two["Asnain"] = "Sidhiquee";

$name\_two["Malhar"] = "Joshi";

// Accessing the elements directly

echo "Accessing the elements directly:\n";

echo $name\_two["Pavan"], "<br>";

echo $name\_two["Malhar"], "<br>";

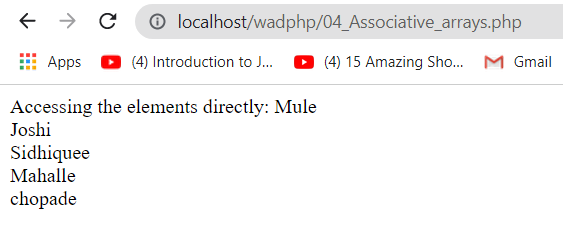
echo $name\_two["Asnain"], "<br>";

echo $name\_one["Rohan"], "<br>";

echo $name\_one["Harish"], "<br>";

?>

**Output:**



1. **Multidimensional Arrays:**

**Theory:** Multi-dimensional arrays are such arrays that store another array at each index instead of a single element. In other words, we can define multi-dimensional arrays as an array of arrays. As the name suggests, every element in this array can be an array and they can also hold other sub-arrays within. Arrays or sub-arrays in multidimensional arrays can be accessed using multiple dimensions.

**Program:**

<?php

// Defining a multidimensional array

$favorites = array(

    "Dave Punk" => array(

        "mob" => "5689741523",

        "email" => "rohan28@gmail.com",

    ),

    "Dave Punk" => array(

        "mob" => "2584369721",

        "email" => "ujjwal@gmail.com",

    ),

    "John Flinch" => array(

        "mob" => "9875147536",

        "email" => "dhiraj@gmail.com",

    )

);

// Using for and foreach in nested form

$keys = array\_keys($favorites);

for($i = 0; $i < count($favorites); $i++) {

    echo $keys[$i] . "\n";

    foreach($favorites[$keys[$i]] as $key => $value) {

        echo $key . " : " . $value . "\n";

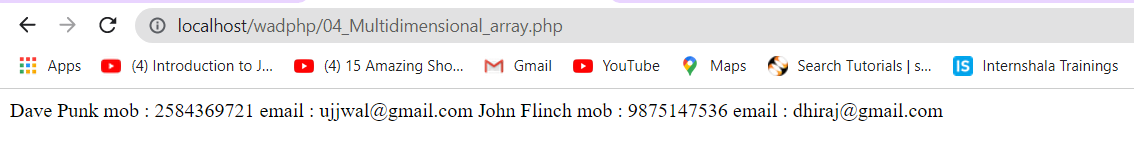
    }

    echo "\n";

}

?>

**Output:**



**Conclusion:** Hence, we have gone through the various type arrays in Php and performed simple tasks.