# PHP Arrays

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

- **Numeric array** − An array with a numeric index. Values are stored and accessed in linear fashion.

- **Associative array** − An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.

- **Multidimensional array** − An array containing one or more arrays and values are accessed using multiple indices

**NOTE** − Built-in array functions is given in function reference PHP Array Functions

## Numeric Array

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

Example

Following is the example showing how to create and access numeric arrays.

Here we have used **array()** function to create array. This function is explained in function reference.

```php
<html>
   <body>

      <?php
         /* First method to create array. */
         $numbers = array( 1, 2, 3, 4, 5);

         foreach( $numbers as $value ) {
            echo "Value is $value <br />";
         }

         /* Second method to create array. */
         $numbers[0] = "one";
         $numbers[1] = "two";
         $numbers[2] = "three";
         $numbers[3] = "four";
         $numbers[4] = "five";
```

```
        foreach( $numbers as $value ) {
            echo "Value is $value <br />";
        }
    ?>

    </body>
</html>
```

This will produce the following result −

```
Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
Value is one
Value is two
Value is three
Value is four
Value is five
```

## Associative Arrays

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employees names as the keys in our associative array, and the value would be their respective salary.

**NOTE** − Don't keep associative array inside double quote while printing otherwise it would not return any value.

Example

```
<html>
    <body>

        <?php
            /* First method to associate create array. */
            $salaries = array("mohammad" => 2000, "qadir" => 1000,
"zara" => 500);

            echo "Salary of mohammad is ". $salaries['mohammad'] .
"<br />";
            echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
            echo "Salary of zara is ".  $salaries['zara']. "<br />";
```

```
        /* Second method to create array. */
        $salaries['mohammad'] = "high";
        $salaries['qadir'] = "medium";
        $salaries['zara'] = "low";

        echo "Salary of mohammad is ". $salaries['mohammad'] .
"<br />";
        echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
        echo "Salary of zara is ".  $salaries['zara']. "<br />";
    ?>

   </body>
</html>
```

This will produce the following result −

```
Salary of mohammad is 2000
Salary of qadir is 1000
Salary of zara is 500
Salary of mohammad is high
Salary of qadir is medium
Salary of zara is low
```

# Multidimensional Arrays

A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

## Example

In this example we create a two dimensional array to store marks of three students in three subjects −

This example is an associative array, you can create numeric array in the same fashion.

```
<html>
   <body>

      <?php
         $marks = array(
            "mohammad" => array (
               "physics" => 35,
               "maths" => 30,
               "chemistry" => 39
            ),

            "qadir" => array (
```

```php
                "physics" => 30,
                "maths" => 32,
                "chemistry" => 29
            ),

            "zara" => array (
                "physics" => 31,
                "maths" => 22,
                "chemistry" => 39
            )
        );

        /* Accessing multi-dimensional array values */
        echo "Marks for mohammad in physics : " ;
        echo $marks['mohammad']['physics'] . "<br />";

        echo "Marks for qadir in maths : ";
        echo $marks['qadir']['maths'] . "<br />";

        echo "Marks for zara in chemistry : " ;
        echo $marks['zara']['chemistry'] . "<br />";
    ?>


    </body>
</html>
```

This will produce the following result −

```
Marks for mohammad in physics : 35
Marks for qadir in maths : 32
Marks for zara in chemistry : 39
```