

PHP 5 Arrays

What is an Array?

Arrays are complex variables that stores multiple or a group of values under a single variable name. Let's suppose you want to store colors in your PHP script.

Storing the colors in single variables could look like this:

একটা ভেরিয়েবল একটা single value এর পাত্র(container). কিন্তু অ্যারে একাধিক value'র container বা পাত্র। একটা অ্যারে হল কিছু উপাদান নিয়ে গঠিত যেখানে প্রত্যেকটি উপাদানের নির্দিষ্ট value আছে-key বা index বলে যেটা দিয়ে উপাদানটিকে রেফার করা হয়। ধরুন আপনার কাছে কিছু শহরের নামের লিস্ট আছে এখন যদি এগুলোকে কোনো single variable এ store করে রাখতে চান তাহলে এমন হবে-

```
<?php
```

```
$city1= "Dhaka"
```

```
$city2= "Chittagong"
```

```
$city3= "Rajshahi"
```

```
$city4= "Sylet"
```

```
$city5= "Khulna"
```

```
$city6= "Barishal"
```

```
?>
```

কিন্তু একটা অ্যারে এসবগুলোকে একটা ভ্যারিয়েবল এ ধরে রাখতে পারে। অ্যারের সাধারণ গঠন পদ্ধতি হচ্ছে কিছু উপাদানের ক্রম (series of element) যার উপাদানগুলির index ০ থেকে শুরু হয়ে ক্রমানুসারে বাড়তে থাকবে। বেশ কয়েকভাবে অ্যারে লেখা যায় সবচেয়ে সহজটি হচ্ছে-

```
<?php
```

```
$city = array("Dhaka", "Chittagong", "Rajshahi", "Sylet", "Khulna", "Barishal");
```

```
Echo $city[2];
```

?>

তবে যদি আমরা চাই তাহলে অ্যারের এই ডিফল্ট ভ্যালু পরিবর্তন করে দিতে পারি, => এই অপারেটরটি দিয়ে। আমি চাচ্ছি এই উপরের অ্যারেটির starting index 2 দিয়ে শুরু হোক তাহলে এভাবে লিখতে হবে।

```
<?php
```

```
$city=array(2=>"Dhaka", "Chittagong", "Rajshahi",  
"Sylet", "Khulna", "Barishal", "Barishal");
```

?>

এই অপারেটর যেকোন উপাদানের পূর্বে ব্যবহৃত হতে পারে তার index নির্দিষ্ট করার জন্য।

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.

Indexed Arrays

An indexed or numeric array stores each array element with a numeric index. The following examples shows two ways of creating an indexed array, the easiest way is:

```
<?php
```

```
$colors = array("Red", "Green", "Blue");
```

```
// Printing array structure
```

```
print_r($colors);
```

?>

Or

```
<?php
```

```
$cars = array("Volvo", "BMW", "Toyota");
```

```
$arrlength = count($cars);
```

```
for($x = 0; $x < $arrlength; $x++) {
```

```
    echo $cars[$x];
```

```
    echo "<br>";
```

```
}
```

```
?>
```

Note: In an indexed or numeric array, the indexes are automatically assigned and start with 0, and the values can be any data type.

Associative Arrays

In an associative array, the keys assigned to values can be arbitrary and user defined strings. In the following example the array uses keys instead of index numbers:

```
<?php
```

```
$ages = array("Peter"=>22, "Clark"=>32, "John"=>28);
```

```
// Printing array structure
```

```
print_r($ages);
```

```
?>
```

Or

```
<?php
```

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

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. An example of a multidimensional array will look something like this:

```
<?php

$continents=array("Asia"=>array("Bangladesh",
"India","Pakistan"),"Europe"=>array("England",
"France"),"Africa"=>array("Kenya","Libya","Somalia"));

print ($continents["Africa"][2]);

?>
```

array_unshift() ফাংশন

একটা অ্যারের সামনে একটা এলিমেন্ট যোগ করতে এই ফাংশন ব্যবহার হয়। যেমন সবার প্রথমে \$city ভেরিয়েবলে যে অ্যারেটি রেখেছি সেই অ্যারের প্রথমে Rangpur এবং Kurigram এই দুটি এলিমেন্ট যোগ করব।

```
<?php

$city = array("Dhaka", "Chittagong", "Rajshahi","Sylet",
"Khulna", "Barishal");

array_unshift($city,"Rangpur","Kurigram");
```

```
/Now $city will be
```

```
$city = array("Rangpur","Kurigram","Dhaka", "Chittagong",  
"Rajshahi","Sylet", "Khulna", "Barishal");
```

```
?>
```

array_push() ফাংশন

একটা অ্যারের শেষে এলিমেন্ট যোগ করতে চাইলে এই ফাংশন ব্যবহার করতে পারেন। যেমন উপরের কোডে `array_unshift` এর জায়গায় শুধু `array_push` যোগ করলেই Rangpur এবং Kurigram এলিমেন্ট অ্যারের শেষে গিয়ে যোগ হবে

array_shift() ফাংশন

এই ফাংশন অ্যারের প্রথম এলিমেন্ট কে মুছে দেবে যদি `$city` অ্যারের প্রথম এলিমেন্ট সরাতে চাই তাহলে

```
<?php
```

```
$city = array("Dhaka", "Chittagong", "Rajshahi","Sylet", "Khulna", "Barishal");
```

```
array_shift($city);
```

```
//Now $city will be
```

```
$city = array("Chittagong", "Rajshahi","Sylet", "Khulna", "Barishal");
```

```
?>
```

array_pop() ফাংশন

এই ফাংশনটি উপরের ফাংশনটির মতই শুধু পার্থক্য হচ্ছে সে অ্যারের শেষ এলিমেন্টটি সরাবে এবং রিটার্ন করবে। উপরের কোডের `array_shift` এর জায়গায় `array_pop` দিয়ে অনুশীলন করে দেখতে পারেন।

in_array() ফাংশন

এই ফাংশন দিয়ে একটা অ্যারের ভিতরে কোন এলিমেন্ট কে খুজে বের করা যায়। যদি খুজে পায় তাহলে **true** রিটার্ন করবে তা নাহলে **false** রিটার্ন করবে। যেমন

```
<?php

$newCity = "Chittagong";

$city = array("Dhaka", "Chittagong", "Rajshahi", "Sylet",
"Khulna", "Barishal");

if(in_array($newCity,$city))

echo "In Bangladesh most of the islamic scholars lives
in $newCity";

?>
```

sort() ফাংশন

এই ফাংশন দিয়ে অ্যারে কে নিম্নমান থেকে উচ্চমান এই ক্রমানুসারে সাজানো যায়। যেমন:

```
<?php

$num = array(2,6,8,1,6,8);

sort($num);

?>
```

- `sort()` - sort arrays in ascending order
- `rsort()` - sort arrays in descending order

PHP asort() Function

The `asort()` function sorts an associative array in ascending order, according to the value.

Tip: Use the [arsort\(\)](#) function to sort an associative array in descending order, according to the value.

Tip: Use the [ksort\(\)](#) function to sort an associative array in ascending order, according to the key.

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
asort($age);

foreach($age as $x=>$x_value)
{
    echo "Key=". $x .", Value=". $x_value;
    echo "<br>";
}
?>
```

PHP ksort() Function

Definition and Usage

The `ksort()` function sorts an associative array in ascending order, according to the key.

Tip: Use the [krsort\(\)](#) function to sort an associative array in descending order, according to the key.

Tip: Use the [asort\(\)](#) function to sort an associative array in ascending order, according to the value.

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
```

```
ksort($age);

foreach($age as $x=>$x_value)
{
    echo "Key=".$x .", Value=".$x_value;
    echo "<br>";
}
?>
```

- arsort() - sort associative arrays in descending order, according to the value
- krsort() - sort associative arrays in descending order, according to the key

PHP array_merge() Function

Definition and Usage

The array_merge() function merges one or more arrays into one array.

Tip: You can assign one array to the function, or as many as you like.

Note: If two or more array elements have the same key, the last one overrides the others.

```
<?php
$a1=array("red","green");
$a2=array("blue","yellow");
print_r(array_merge($a1,$a2));
?>
```

PHP array_slice() Function

Definition and Usage

The array_slice() function returns selected parts of an array.

```
<?php
$a=array("red","green","blue","yellow","brown");
```



```
print_r(array_slice($a,2));  
?>
```

PHP array_splice() Function

Definition and Usage

The `array_splice()` function removes selected elements from an array and replaces it with new elements. The function also returns an array with the removed elements.

Tip: If the function does not remove any elements (`length=0`), the replaced array will be inserted from the position of the `start` parameter (See Example 2).

Note: The keys in the replaced array are not preserved.

```
<?php  
$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");  
$a2=array("a"=>"purple","b"=>"orange");  
array_splice($a1,0,2,$a2);  
print_r($a1);  
?>
```

PHP in_array() Function

Definition and Usage

The `in_array()` function searches an array for a specific value.

Note: If the search parameter is a string and the type parameter is set to `TRUE`, the search is case-sensitive.

```
<?php  
$people = array("Peter", "Joe", "Glenn", "Cleveland");
```

```
if (in_array("Glenn", $people))
{
    echo "Match found";
}
else
{
    echo "Match not found";
}
?>
```

PHP 5

Total Array Functions.....follow this
link.....

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

<http://php.net/manual/en/ref.array.php>