

Name : Mayur Purushvani

Introduction of Array :

Array is an collection of elements.

Which stores different elements in single array variable.

We can perform various operations in array like push, pop, remove, etc.

Example :

```
<?php
$arr = array('Bike', 'car', 'rixo', 'cycle');
print_r( $arr );

echo "<br>".$arr[1]."<br>";

$arr[4] = 'scooty';

print_r ( $arr );

?>
```

Associative Arrays :

An associative array is also called ad dictionary.

In dictionary we have a key value pairs.

So, In this concept there is nothing changed, like we can manually add a key value pair in the array

Example :

```
<?php
$arr = array('Bike'=>20, 'car'=>50, 'rixo'=>70, 'cycle'=>100);
print_r( $arr );

//echo $arr[0]; //It Will not work now.

echo $arr['Bike'];

?>
```

Multi-dimensional Arrays :

A multi-dimensional array are extension of 2-D arrays, 3-D arrays, etc...

An Array inside array is called multi-dimensional arrays.

Example :

```
<?php

$food = array('Healthy' =>
              array('vegetable','salad'),
              'Unhealthy' =>
              array('pizza', 'Ice Cream'));

echo $food['Healthy'][1];

echo $food['Unhealthy'][1];

?>
```

For-each loop :

The foreach loop works only in arrays and used to loop through each key/value pair in an array.

Example :

```
<?php

$food = array('Healthy' =>
              array('vegetable','salad'),
              'Unhealthy' =>
              array('pizza', 'Ice Cream'));

foreach ( $food as $elements => $items_array)
{
    echo '<b>'.$elements.'</b><br>';
    foreach( $items_array as $items_names)
    {
        echo $items_names.'<br>';
    }
}

?>
```

Include and require function :

The include and require functions is used to include the file into another file.

The basic difference between this 2 is when we write include statement with The file does not exists, It will display the below default code also.

But when we write require statement with the file does not exists, It will not execute below code. And not display the default code in web page.

Like below example :

Take a “Hello This is” statement, and try with both the functions require and include.

In include you can see that statement in web page and in require you cannot see that.

```
<?php

//include ('header.php');

//require ('header.php');

//include ('hhh.php'); /try this one
//require ('hhh.php'); //try This one

echo "Hello! This is ".$val1;

?>
```

The syntax of include and require is :

You can also give round braces if you want or else it will work fine.

Include (“header.php”); OR include ‘header.php’;

Include_once and require_once :

We’ve seen the basic difference between the require and include.

The include statement include the file while that file doesn’t exists.

And require must require that file to exists.

Now, the require_once and require is little different, When we write require function, Then write require_once function, The output will be display twice.

The `require_once` function checking that file into a directory and if not exists, It will display an error.

Example :

```
<?php

require ('header.php');

//require_once('header.php'); //Now cut this line and put it above the
require. It will execute twice.

/*
if(defined('header.php')) {
    require 'header.php'
}
*/
//This above condition is same work as require_once. Now you can see th
e basic difference and it is same as include and include_once.
?>
```

Expression Matching :

Previously we've used `preg_match()` function to match the string in the original string.

Here is `has_space()`.

Example :

```
<?php

function has_space($string){
    if(preg_match('/ /', $string)) {
        return true;
    }
    else {
        return false;
    }
}

$string = "Thisismayurformcybercomcreation!";

if(has_space($string)){
    echo 'Space is between!';
}
else {
    echo 'No space';
}
?>
```

Strlen function :

Example:

```
<?php

$string = 'Mayur';

$string_length = strlen($string);

for($x = 1; $x <= $string_length; $x++){
    echo $x."<br>";
}

?>
```

String uppercase and lowercase example :

```
<?php

if(isset($_GET['username']) && !empty($_GET['username']))
{
    $username = $_GET['username'];
    $username_lc = strtolower($username);
    if($username_lc == 'mayur') {
        echo 'all the best!';
    }
}

?>

<form action="upper_lowercasefunctions.php" method="get">
    Name : <input type="text" name="username"><br><br>
    <input type="submit" value="OK">
</form>
```

Strpos :

This function has 3 arguments, first is string itself, second is the string that we've to search and the third is offset (optional). Offset is from where we have to search the string.

It will return only the first search string.

Exmple :

```
<?php
$find = "mayur";
$find_length = strlen($find);
$string = 'This is mayur purushvani. I\'m mayur. My name is mayur';
echo strpos($string, $find, 35);
?>
```

With multiple search :

```
<?php
$find = "mayur";
$offset = 0;
$find_length = strlen($find);
$string = 'This is mayur purushvani. I\'m mayur. My name is mayur';
//echo strpos($string, $find, 35);

while ($string_position = strpos($string, $find, $offset)) {
    echo '<strong>'. $find.'</strong> found at '.$string_position.'<br>';
    $offset = $string_position + $find_length;
}
?>
```

Substr_replace():

This function has 4 arguments.

First is string itself.

Second is the replaced string.

Third is where we want to start to replace and forth is where from howmany characters do you want to end the replace.

Example :

```
<?php
$string = "Hello guys! This is mayur purushvani";
$string_replace = substr_replace($string, "max", 20, 5);
echo $string_replace;
?>
```