

int array\_push (array &\$array, mixed \$var[, mixed \$...])

It treats array as a stack, pushes the passed variables onto the end of array. It returns the new total number of elements in the array.

Example:

```
<?php
$stack = array("orange", "banana");
array_push($stack, "apple", "raspberry");
print_r($stack);
```

?>

Output:

Array

{

[0] => orange

[1] => banana

[2] => apple

[3] => raspberry

}

**str ini-get (string \$varname)**

Returns the value of the configuration option on success otherwise null value or empty string.

Example:

```
<?php
echo ini-get('upload_max_filesize');
?>
```

Output:

2M

**array ini-get-all()**

Returns all configuration options in the form of associated array.

Date  
02/02/2017

Short-open-tag [make it On in php.ini to make use of php short<sup>open</sup> tag <? instead of <?php]

<?php echo \$a ?> is equal to <?= \$a ?>

If a file is pure PHP code, it is preferable to omit the PHP closing tag at the end of the file. This prevents accidental whitespace.

Short open tag is discouraged since it is available only if enabled in php.ini.



### About [mxtoolbox.com](https://mxtoolbox.com)

Checking that an email is coming from a secure email server or not.

Example: Go to [jiten@research@gmail.com](mailto:jiten@research@gmail.com) and search with 'mxtoolbox.com'

1. Get coming mail & click on 'show original'
2. Take copy of mail header
3. Go to [mxtoolbox.com](https://mxtoolbox.com) & click on 'Analyze Headers'
4. Taking from step 2, paste in text area & click on 'Analyze Header'.
5. Check your mail is coming through your used email server or not. In this case, I am using an email server 'secureserver.net'

## array\_search()

Searches the array for a given value and returns the first corresponding key if successful; FALSE otherwise.

### Example:

```
$array = array(0 => 'blue', 1 => 'red', 2 => 'green',  
               3 => 'red');
```

```
$key = array_search('green', $array); // $key = 2
```

```
$key = array_search('red', $array); // $key = 1
```

## array\_combine()

Creates an array by using one array for keys another for its values.

Return the combined array, FALSE if the number of elements for each array isn't equal.

Example:

<?php

```
$a = array('green', 'red', 'yellow');
```

```
$b = array('avocado', 'apple', 'banana');
```

```
$c = array_combine($a, $b);
```

```
print_r($c);
```

```
?>
```

Output:

Array

(

[green] => avocado

[red] => apple

[yellow] => banana

)