

Adding Elements in an Array

In this lesson, we will learn about adding elements at the start, end and in between an array

We'll cover the following

- Adding Elements at the Start
 - Using `array_unshift()`
- Adding Elements at the End
 - Using `array_push()`
- Adding and Replacing Values at Random Position in an Array

Adding Elements at the Start

Sometimes you want to add an element to the beginning of an array without modifying any of the current elements within the array. Whenever this is the case, you can use `array_unshift()` function.

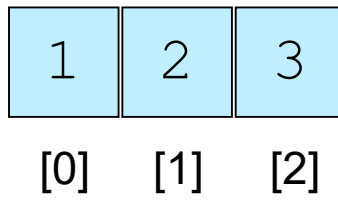
`array_unshift()` prepends passed elements at the front of an array. Note that the list of elements is prepended as a whole so that the prepended elements stay in the same order. All numerical array keys **will be modified** to start counting from 0.

Using `array_unshift()`

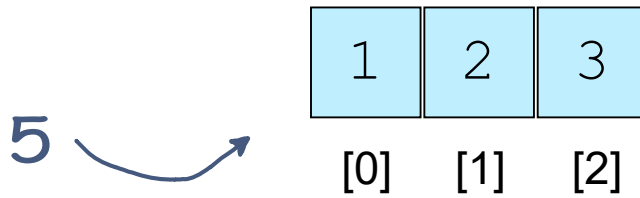
`array_unshift()` has **two or more** input parameters: the first parameter is the name of the array, and the subsequent parameters are the elements to be added at the start of that array.

Following is an example for adding elements at the start of an array:

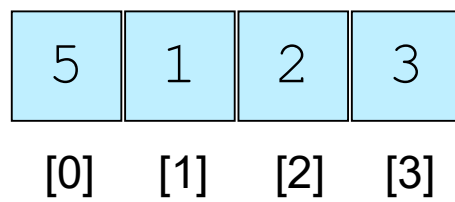
```
array_unshift($myarray, 4, 5)
```



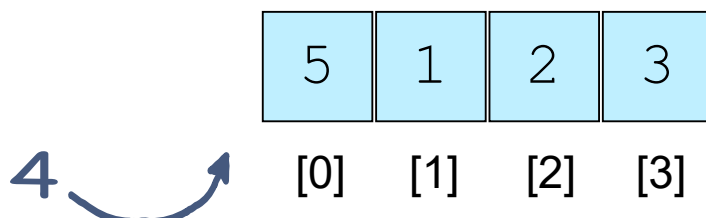
1 of 5

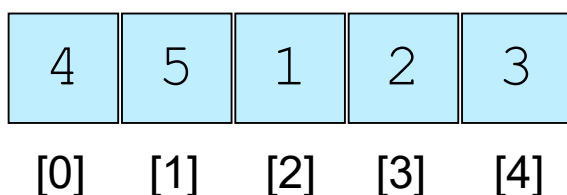


2 of 5



3 of 5





Run the following code to see how this works:

```
<?php
$myArray = array(1, 2, 3);
array_unshift($myArray, 4,5);
print_r($myArray)
?>
```



Adding Elements at the End

Sometimes you want to add an element at the end of an array without modifying any of the current elements within the array. Whenever this is the case, you can use `array_push()` function.

`array_push()` function pushes one or more elements at the end of an array. All numerical array keys will remain same and new ones will be created corresponding to the new elements pushed.

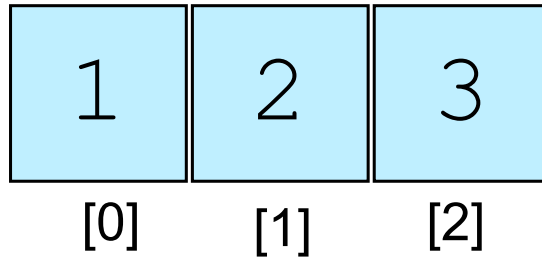
Using `array_push()`

`array_push()` also has **two or more** input parameters: the first parameter is the name of the array, and the subsequent parameters are the elements to be added at

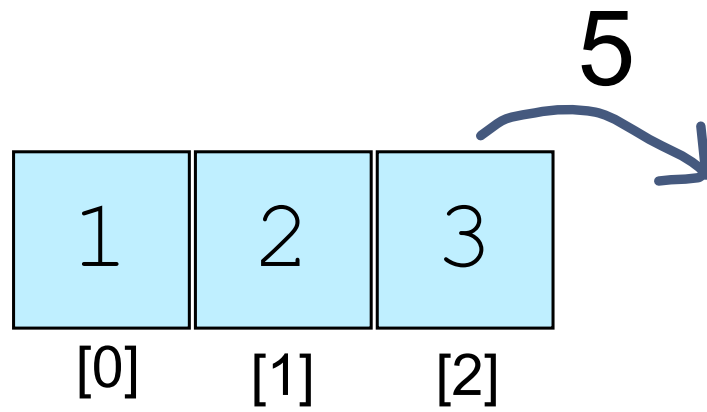
end of that array.

Following is an example for adding elements at the end of an array:

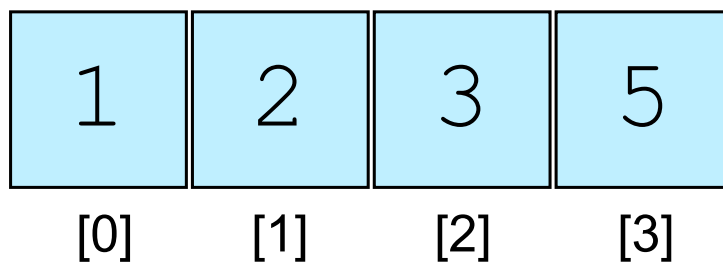
```
array_push($myarray, 5, 6)
```



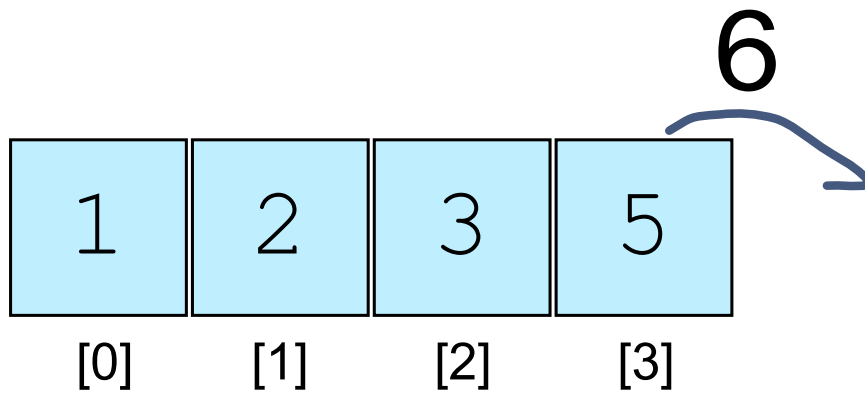
1 of 5



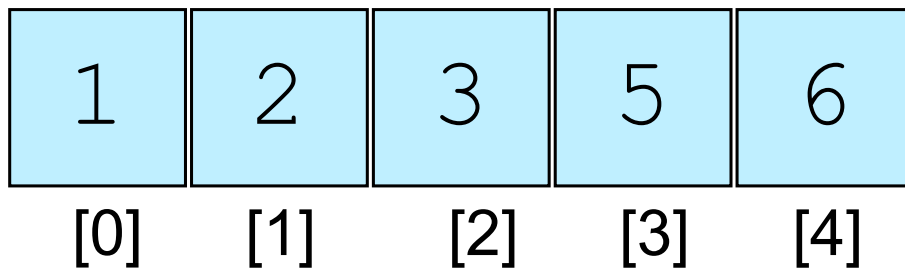
2 of 5



3 of 5



4 of 5



5 of 5



Run the code snippet below to see how this is done in PHP:

```
<?php
$array = [1,2,3];
print_r($array);
array_push($array, 5, 6); // Pushing 5 and 6 at the end of $array
print_r($array);
?>
```



Adding and Replacing Values at Random Position in an Array

To add or replace values at any position in an array, access the array position using the key and assign a value. This is shown as below:

```
$arrayName[key]=value;
```

- If an old key is used to assign a new value, the old value will be replaced.
- If a new key is used to assign a value, a new key will be created in the array.

The following code widget shows how this is done in PHP:

```
<?php
$arr=array(1,3,5,7,9);
$arr[23]=71; // adding a new key and its associated value
print_r($arr);

$arr[2]=22; // replacing an old value at 2nd index
print_r($arr);
```



In the next lesson, we will discuss another array operation – removing elements in array.